

Design Of Accurate And Smooth Filters For Function And Derivative Reconstruction

Torsten Möller^{1,2}, Klaus Mueller¹, Yair Kurzion¹, Raghu Machiraju^{3,4}, Roni Yagel^{1,2}

¹Department Of Computer And Information Science

²The Advanced Computing Center For The Arts And Design
The Ohio State University, Columbus, Ohio

³NSF Engineering Research Center For Computational Field Simulation

⁴Department of Computer Science
Mississippi State University, Mississippi

ABSTRACT

The correct choice of function and derivative reconstruction filters is paramount to obtaining highly accurate renderings. Most filter choices are limited to a set of commonly used functions, and the visualization practitioner has so far no way to state his preferences in a convenient fashion. Much work has been done towards the design and specification of filters using frequency based methods. However, for visualization algorithms it is more natural to specify a filter in terms of the smoothness of the resulting reconstructed function and the spatial reconstruction error. Hence, in this paper, we present a methodology for designing filters based on spatial smoothness and accuracy criteria. We first state our design criteria and then provide an example of a filter design exercise. We also use the filters so designed for volume rendering of sampled data sets and a synthetic test function. We demonstrate that our results compare favorably with existing methods.

Keywords: Interpolation (G.1.1) Approximation (G.1.2) Quadrature and Numerical Differentiation (G.1.4) Picture/Image Generation (I.3.3) Reconstruction (I.4.5)

Other Keywords: Volume Rendering, Filter Design, interpolation, derivatives

1. INTRODUCTION

The reconstruction of a function and its derivatives from a set of given samples of that function is a fundamental operation in many areas. Computer graphics, scientific visualization, and image processing are just a few examples. In all these areas, a set of samples of an unknown function is usually all we know of that function. Hence, the reconstruction of the function between sample points is

rather arbitrary, and one cannot talk about an *ideal* reconstruction function. The notion of *ideal* reconstruction is based on the assumption, that a given function is a member of a certain functional space, e.g. the L^2 space or the bandlimited function space. This functional space (and therefore the *ideal* reconstruction method) is usually determined by the particular application. In visualization, and in other fields, we assure that the given function belongs to the space of smooth functions C^n , where n is an integer.

A very important and often studied space in the class of all smooth functional spaces, is the space of bandlimited functions (a subclass of C^∞). They are often studied in the frequency domain using a signal processing approach. Although these methods are capable of controlling *global* errors such as blurring and aliasing, no *local* spatial assessment of their accuracy can be conducted directly. It turns out that the ideal reconstruction filters for the space of bandlimited functions are impractical to use. Hence research in this area has focused on finding efficient filters that approximate the ideal filter [1][4][7][9][10][14][15].

Another body of work has concentrated on minimizing the local spatial error for design and evaluation of filters [11][16][17][20]. The local error was measured and minimized using a Taylor series expansion. Since visual perception, judged by ringing, aliasing and blurring, was of concern, the frequency behavior of the resulting filter was discussed. In addition, spatial design gives an easy control over the size of the filter, and hence on the efficiency of the resulting filter, a property frequency-based methods do not have. However, it was found that the sole concern for numerical accuracy can lead to discontinuous filters, which can produce visual artifacts that are easily detected [17]. The goal of this paper is to overcome this problem by introducing a smoothness requirement into the filter design process.

All filter designs in the spatial domain have built filters according to an accuracy criteria. In this paper we introduce, for the first time, filter design criteria for interpolation and derivative filters which yield functions with a minimal numerical error and still maintain good spectral properties. The only assumption that we require of the original function (represented by the given samples) is that it is smooth and a member of the functional space C^n . We also show how our design criteria relate to criteria in frequency domain.

Our filter design is not restricted to cubic polynomial basis functions, but can generate filters of arbitrary smoothness and accu-

0-8186-9180-8/98/\$10.00 Copyright 1998 IEEE

racy. In this paper, we design optimized piecewise polynomial interpolation filters according to a set of smoothness and accuracy requirements. These filters are drawn from the set of all piecewise polynomial filters, a more general class of filters than the popular BC-splines. Since our methods also apply easily to any derivative filter design, we find optimal piecewise polynomial gradient filters as well. Our results, which go beyond the de-facto standard of the popular cubic BC-splines introduced by Mitchell and Netravali [16], are summarized in Table 1 and Table 2. These tables provide a guide to which filters should be used in most applications. Furthermore, we provide the practitioner with an easy and fast way to design filters that are specific to their applications by determining an application oriented set of smoothness and accuracy criteria.

The outline of this paper is as follows; Section 2 summarizes previous research in this field. In Section 3, we introduce the design criteria that we use in Section 4 to design new filters. In Section 5 we present some experimental results and in Section 6 we suggest steps for furthering this research. Finally, in Section 7, we summarize our findings.

2. PREVIOUS RESEARCH

Two of the more important and well studied reconstruction algorithms are interpolation and gradient estimation. In volume rendering, we must be able to interpolate the function at arbitrary locations to obtain the volume densities needed for arbitrary viewing. The gradient (the first derivative of the function) is employed in both volume classification and shading [6][13]. If the gradient estimation is done incorrectly, shading and classification will yield misleading colors and opacities.

Many researchers have shown that the *Sinc* filter is an ideal interpolation filter for the space of bandlimited functions (a subclass of C^∞). In this space the *Cosc* filter, which is the analytic derivative of the *Sinc* filter, is an ideal derivative filter [1][7][19]. These filters completely cut off the frequencies above a certain Nyquist frequency. Because of this discontinuity in the frequency domain, those filters have infinite support in the spatial domain and therefore are impractical to use for digital signals. Windowing the *Sinc* filter was introduced in order to smoothly limit this filter spatially [10][19]. Carlbom [4] computed an approximation to a modified *Sinc* filter with a minimized Chebychev error. Goss [9] extended the idea of windowing from interpolation filters to derivative filters. He used a Kaiser window to mitigate the adverse effects of the truncated ideal derivative filter. Instead of trying to find a good approximation to the ideal filter for all frequency ranges, Dutta Roy and Kumar's filter design [7] can be easily adapted to find good approximations for select frequency ranges.

A comparative study by Marschner and Lobb [15] proposed the use of different error metrics for various reconstruction artifacts of interpolation filters. These error metrics operate in the frequency domain and measure the smoothing, post-aliasing, and overshoot attributes of an interpolation filter. This study showed that the windowed *Sinc* filter has the best behavior.

In the spatial domain, Keys [11] analyzed a certain class of cubic splines, also called *cardinal splines*, using a Taylor series expansion. He showed that, within this class, the Catmull-Rom spline is optimal in the sense that it interpolates the original function with the smallest asymptotic spatial error. He also graphically compared the Catmull-Rom spline with the ideal interpolation filter, noticing

that it is suitable for practical applications in computer graphics. Mitchell and Netravali [16] introduced a more general class of cubic splines which we refer to as *BC cubic splines* or in short, *BC-splines*. Cardinal cubic splines are a subclass of the BC-splines. Mitchell and Netravali conducted a study involving more than 500 sample images, classifying the parameter space into different regions of dominating reconstruction artifacts such as blurring, ringing, and anisotropy. They found, by using a Taylor series expansion, that filters for which $B + 2C = 1$ are the most numerically accurate within the class of BC-splines and have an error proportional to the square of the sampling distance. They also found, through their empirical studies, that these filters, although numerically superior, are not always visually superior.

Recently, we have shown [17] that the derivative approximation has a larger impact on the quality of the volume rendered image than the interpolation operation and therefore deserves a thorough analysis. Unfortunately, not much work has been done in the spatial design of derivative filters. Bentum et al. [1] use the Cardinal cubic splines as a basis to constructing the derivative filter through an analytic derivation of the interpolation filter. Although the authors illustrate the effect of various parameters on these filters via a number of frequency plots, they do not analytically compare the different filters. We (in [17]) have developed tools for the spatial analysis of both interpolation and derivative filters of arbitrary order. We used a Taylor series of the convolution sum in order to come up with four evaluation criteria. These criteria include asymptotic, as well as absolute, local error effects of the filter on the reconstructed function. We use these criteria in our current paper as a way to control the numerical error of the filters that we design. Using the methods developed in [17] we conducted a comparison of various derivative (normal vector) reconstruction methods and classified them into four reconstruction schemes [18].

Since we will employ the results of [17] throughout our paper, we include a summary here:

2.1 Taylor Expansion of the Convolution Sum

To reconstruct a continuous function $f(t)$ or its derivative $f'(t)$ from a set of sample points $f[k]$, we convolve $f[k]$ with a continuous filter kernel w . The filter w can be either an interpolation or a derivative filter. We denote the result of this operation by $f_r^w(t)$. Formally, this can be written as:

$$f_r^w(t) = \sum_{k=-\infty}^{\infty} f[k] \cdot w\left(\frac{t}{T} - k\right), \quad (1)$$

where T is the sampling distance. Now we can expand $f[k] = f(kT)$ into a Taylor series of $N+1$ terms about t . The Taylor series expansion at that point would be:

$$f[k] = \sum_{n=0}^N \frac{f^{(n)}(t)}{n!} (kT - t)^n + \frac{f^{(N+1)}(\xi_k)}{(N+1)!} (kT - t)^{(N+1)}$$

where $f^{(n)}(t)$ is the n -th derivative of f and $\xi_k \in [t, kT]$.

Substituting the Taylor series expansion into the convolution sum of Equation 1, leads to an alternative representation for the reconstructed value at a point t :

$$\begin{aligned}
f_r^{(v)}(t) &= \sum_{n=0}^N a_n^w(\tau) f^{(n)}(t) + r_{N,i}^w(\tau) \\
a_n^w(\tau) &= \frac{T^n}{n!} \sum_{k=-\infty}^{\infty} (k-\tau)^n w(\tau-k) \\
r_{N,i}^w(\tau) &\leq \left(\max_{\xi \in [(i-M)T, (i+M)T]} (f^{(N+1)}(\xi)) \right) |a_{N+1}^w(\tau)| \\
&\text{or} \\
r_{N,i}^w(\tau) &\approx a_{N+1}^w(\tau) f^{(N+1)}(t)
\end{aligned} \tag{2}$$

where τ is chosen such that $t = (i + \tau)T$, with $0 \leq \tau < 1$, and i is an integer. It is noteworthy that the derived error coefficients a only depend on the offset τ to the nearest sampling point, i.e., they are periodic in the sampling distance T . For further details, please refer to [17].

The characterization of the filtering process in Equation 2 imposes four different criteria for a good reconstruction scheme of the k -th derivative. First of all, we require a_n^w to be zero for all n smaller than k . Secondly we have to normalize by a_k^w in order to reconstruct the actual derivative as opposed to some multiple of it. Further by determining the largest N , such that a_N^w is zero, we can determine the asymptotic error behavior of a filter for a decreasing sampling distance T . Finally, the remainder term r gives us an indication of the absolute error of that filter.

This expansion of the convolution sum assumes that at least the first N derivatives of the function f exist, where N depends on our error analysis. Hence, we assume that the underlying function is a member of the class of smooth functions C^N . This condition is generally met in practice [2][17][21].

3. DESIGN CRITERIA

Whenever we are trying to reconstruct a function from sample points we are hoping that the reconstruction process performs *well* and we don't get many *artifacts*. However our understanding of such terms like *good reconstruction* as well as *artifacts* during this process is usually highly dependent on the specific application. Most applications share an attempt to recover the original sampled function as accurately as possible. In order to measure the accuracy of the process, one must have an idea about the type of original function from which the samples were recovered. As we have pointed out in Section 2.1, it is not restrictive to most applications to assume that the original function is continuous to some degree n and therefore belongs to the class of functions C^n . This is the only assumption, that we require for our filter design.

In addition of hoping for an accurate function reconstruction, almost all applications will require the reconstruction of a smooth function. Since we assume a smooth original function, it is natural to expect a smooth function as the result of the reconstruction process. A smooth reconstruction will also guarantee the disappearances of image artifacts in visualization and imaging applications. Another application is CAD in which designers reconstruct surfaces from a set of sample points (knots) using basis functions that are developed to yield surfaces of C^1 , C^2 or higher continuity. The reason for smooth function reconstruction is that our visual system is capable of detecting and enhancing even small discontinuities in images. For example, in Fig. 2b and Fig. 4b (see color plates) we

reconstructed a test function (introduced in [15]) and an MRI data set with a filter that was designed solely by requiring high accuracy [17] and therefore yields very little absolute error. It, however, suffers from discontinuities, leading to a discontinuous reconstructed function. Consequently, we would like to reconstruct a function, that is a member of the continuous function space C^n .

For practical applications, the efficiency of the reconstruction process is of great importance as well. In volume rendering, the efficiency of a reconstruction filter, which is employed routinely many times [1], is a source of great concern. It is desirable to use as few samples as possible in order to reconstruct the function at a new location.

We conclude that for general filter design we have to answer three questions:

- What derivative of the original function do we want to reconstruct?
- What accuracy do we require from the reconstruction process?
- What space C^n should the reconstructed function belong to?

Commonly there is also the question of how many filter weights should the filter have. We have elegantly answered this question by minimizing the number of weights and by designing the most efficient filter fulfilling the constraints of the design.

The first two questions can easily be expressed using the framework developed in [17]. Assuming that we want to reconstruct the k -th derivative of the given digital signal f , we simply require that all error coefficients a_n^w in Equation 2 be zero, where $n < k$. Further, we require that the coefficient of the k th derivative be one. Formally, this can be expressed as:

Condition 1: $a_n^w = 0$ for all $n < k$ and $a_k^w = 1$.

The major goal of the design in spatial domain is numerical accuracy. We gain numerical accuracy by requiring the error coefficients a_n^w beyond k to be zero. This leads to what we call N-EF filters (that is, Error Function of the N^{th} order) ([17]):

Condition 2: $a_n^w = 0$ for all $k < n < N + k - 1$.

It is no restriction to consider the filter w to be composed of elements w_k , which are defined by

$$w_k = w_k(\tau) = w(\tau + k).$$

An example of this piecewise decomposition of the filter w is illustrated in Fig. 1. Now it is easy to see that the first two conditions yield an equation system in the unknowns w_k . The solution of that equation system will define a filter w , that fulfills *Condition 1* and *2*. Since this is a linear equation system, we can easily solve it symbolically by Gaussian elimination, which yields a solution for the w_k . This concludes the first step of our function design, defining a class of filters, that guarantee an N-EF accurate reconstruction of the k th derivative of the original function.

As we have pointed out earlier, not every filter of this class yields a continuous reconstructed function and therefore might lead to undesirable artifacts. Hence, we desire the reconstructed function to be part of a smooth function space C^M . From Equation 1 it is clear that we need to require our filter w to be in this class. In order to be a member of C^M , a function w and its M derivatives must all

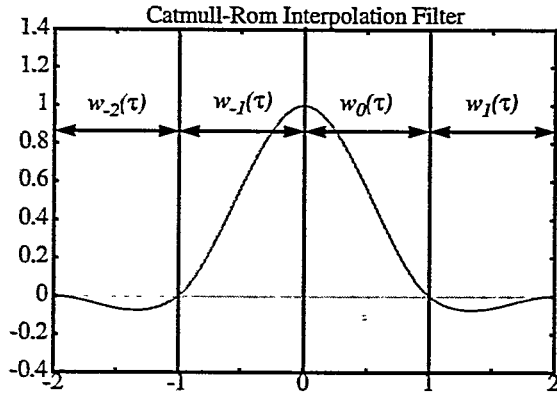


FIGURE 1. A piecewise polynomial interpolation filter using four filter weights $w_k = w_k(\tau) = w(\tau + k)$. Essentially the filter has p parts. For both symmetric and anti-symmetric filters p is even.

be continuous everywhere, including every open interval $(k, k+1)$ for every integer k , and also at all the integer points k themselves. Since the equation system of *Conditions 1* and *2* yields a piecewise filter kernel, we can mathematically express the smoothness criteria as:

Condition 3: $w_k(\tau) \in C^M$ and $w_k^{(m)}(1) = w_{k+1}^{(m)}(0)$ for all k and all $m < M$, where $w_k^{(m)}$ denotes the m -th derivative of w_k .

After solving the equation system of *Conditions 1* and *2* and determining the smoothness of the desired filter, we have a new set of criteria for our filter that needs to be met. In order to design an actual filter, we have to find a solution that fulfills all these conditions. While it is not necessary to restrict oneself to piecewise polynomial filters, we have done so here. The reason for this is that they are easy to use and implement, and are therefore very popular. Now *Conditions 1*, *2* and *3* translate to a linear equation system in the coefficients of piecewise polynomials. The solution of this equation system yields a class of polynomials. These can be further restricted by choosing efficient filters, i.e. with the least number of filter weights and small degrees of polynomials. This concludes the filter design. Summarizing the filter design includes the following steps:

Step 1: Solve a linear equation system created by *Conditions 1* and *2* in the pieces w_k of the filter w .

Step 2: Choose a set of basis functions for the representation of w_k .

Step 3: Solve for the coefficients of the basis functions, considering *Condition 3* as well as the solution of *Step 1*.

Our design criteria also have validity in the frequency domain. It can be shown that our accuracy criteria, defined by specifying the error coefficients a_k^w , translate to conditions on the frequency representation of the filter w at the DC value. a_0^w represents the DC value itself and a_k^w the k -th derivative of the frequency spectra at that point. This is a very desirable condition and was suggested as a filter design criteria by Dutta Roy et al. [7], for designing maximal linear filters. Since it is impractical to use an ideal reconstruction filter (in the C^∞ sense), their idea was to design filters that

come very close to the ideal filter in parts of the frequency spectra and includes some important frequencies. For general applications, we would expect to have the most important frequencies around the DC value.

Since the accuracy criteria only fixes the frequency domain at a single point, it is not enough to guarantee well behaved filters. Our smoothness criterion in *Condition 3* constructs filters w of the class C^M . That means they can be decomposed (using a Taylor series) into a polynomial of M^{th} degree and a remainder term. Now, the polynomial of M^{th} degree translates into a function defined as $\omega^{-(M-1)}$ in frequency space [3]. This guarantees a quick decay of our reconstruction filter. The higher the smoothness condition, the quicker the decay. This ensures that aliasing effects of our designed filter diminish with increasing M .

Having explained the general design process, we turn to demonstrate it by ways of an example.

4. EXAMPLE

Let us assume we want to construct a derivative filter. We expect this derivative filter to be somewhat reliable in terms of accuracy, so we choose a 2EF filter. Further, we aim for a C^1 continuous filter. That leads to three conditions to fulfill:

1. derivative filter:

$$\begin{aligned} a_0(\tau) &= 0 \\ a_1(\tau) &= 1 \end{aligned} \quad (3)$$

2. numerical accuracy 2EF:

$$a_2(\tau) = 0 \quad (4)$$

3. smoothness C^1 :

$$w \in C^1 \quad (5)$$

Here $a_k(\tau)$ are the error coefficients defined in Equation 2 with a positive offset τ , $0 \leq \tau < 1$. The filter to be constructed is w . Decomposing our filter w in pieces w_k as mentioned in Section 3, we can write the three conditions above in terms of the filter weights $w_k = w_k(\tau) = w(\tau + k)$. We will use the notation w_k and $w_k(\tau)$ interchangeably. Using the definition of the error coefficients in Equation 2 will simplify the conditions of Equation 3 through Equation 5 to:

1. derivative filter:

$$\begin{aligned} w_{-2} + w_{-1} + w_0 + w_1 &= 0 \\ T((2-\tau)w_{-2} + (1-\tau)w_{-1} + (-\tau)w_0 + (-1-\tau)w_1) &= 1 \end{aligned} \quad (6)$$

2. numerical accuracy 2EF:

$$(2-\tau)^2 w_{-2} + (1-\tau)^2 w_{-1} + (-\tau)^2 w_0 + (-1-\tau)^2 w_1 = 0 \quad (7)$$

3. smoothness: (here w'_n denotes the derivative of w_n)

$$\begin{aligned}
& w_k \in C^1 \quad \text{and} \\
& w_{-2}(0) = 0 \quad w'_{-2}(0) = 0 \\
& w_{-2}(1) = w_{-1}(0) \quad w'_{-2}(1) = w'_{-1}(0) \\
& w_{-1}(1) = w_0(0) \quad \text{and} \quad w'_{-1}(1) = w'_0(0) \\
& w_0(1) = w_1(0) \quad w'_0(1) = w'_1(0) \\
& w_1(1) = 0 \quad w'_1(1) = 0
\end{aligned} \quad (8)$$

The choice of the number of filter weights w_k (which is the same as the number of piecewise, non-zero parts of the function w) is rather arbitrary. If we choose too many, the resulting filter becomes inefficient. If we choose too few, the equation system might not lead to a solution at all. Since we are trying to design cost-efficient filters, we'd like to have as few as possible filter weights. Since, in computer graphics, we are interested in anti-symmetric derivative filters (symmetric interpolation filters) and the weights $w_k(\tau)$ are defined over integer intervals, we always need an even number of weights (Fig. 1). Conditions 1 and 2 already impose three equations on the filter weights, thus we expect at least four weights to be necessary for our resulting filter.

The equation system in Equation 6 and Equation 7 has three equations in the four unknowns w_k . Therefore it is under-determined and leads to the following set of solutions (setting T to 1):

$$\begin{aligned}
w_1 &= w_1 \\
w_0 &= -3w_1 + \tau - \frac{3}{2} \\
w_{-1} &= 3w_1 - 2\tau + 2 \\
w_{-2} &= -w_1 + \tau - \frac{1}{2}
\end{aligned} \quad (9)$$

Any filter w whose filter weights fulfill Equation 9 is guaranteed to be a 2EF first derivative filter. The actual filter can be constructed using specific basis functions for the w_k and insuring that our smoothness condition (Equation 8) is fulfilled. An obvious choice for the C^1 continuous basis function would be polynomials, since polynomials are a member of C^∞ . Using our notation $w_k = w_k(\tau)$, we require:

$$w_k(\tau) = C_k \tau^2 + B_k \tau + A_k,$$

where the coefficients C_k, B_k, A_k are unknown and remain to be determined. Here again the choice of a second order polynomial is rather arbitrary. If we choose too high of a degree, we get an inefficient, parameter depending solution. For too low of a polynomial degree, we might not get a solution, since not all constraints on the w_k from Equation 9 and Condition 3 can be fulfilled. Substituting this definition of the filter weights w_k into Equation 8 yields the following condition on their coefficients:

$$\begin{aligned}
A_{-2} &= 0 & B_{-2} &= 0 \\
C_{-2} + B_{-2} + A_{-2} &= A_{-1} & 2C_{-2} + B_{-2} &= B_{-1} \\
C_{-1} + B_{-1} + A_{-1} &= A_0 & 2C_{-1} + B_{-1} &= B_0 \\
C_0 + B_0 + A_0 &= A_1 & 2C_0 + B_0 &= B_1 \\
C_1 + B_1 + A_1 &= 0 & 2C_1 + B_1 &= 0
\end{aligned}$$

Substituting the polynomial definition of the filter weights w_k into Equation 9 yields these conditions on their coefficients:

$$\begin{aligned}
C_0 &= -3C_1 & B_0 &= -3B_1 + 1 & A_0 &= -3A_1 - \frac{3}{2} \\
C_{-1} &= 3C_1 & B_{-1} &= 3B_1 - 2 & A_{-1} &= 3A_1 + 2 \\
C_{-2} &= -C_1 & B_{-2} &= -B_1 + 1 & A_{-2} &= -A_1 - \frac{1}{2}
\end{aligned}$$

Further, requiring anti-symmetric filters, yields the following conditions:

$$w_{k-1}(\tau) = -w_k(1-\tau),$$

which translates to

$$\begin{aligned}
C_{k-1} &= -C_k \\
B_{k-1} &= 2C_k + B_k \\
A_{k-1} &= C_k + B_k + A_k
\end{aligned}$$

in all positive k . This leads to an equation system in the coefficients, which solved and substituted into Equation 9, leads to the following filter weights:

$$\begin{aligned}
w_1 &= -\frac{1}{2}\tau^2 + \tau - \frac{1}{2} \\
w_0 &= \frac{3}{2}\tau^2 - 2\tau \\
w_{-1} &= -\frac{3}{2}\tau^2 + \tau + \frac{1}{2} \\
w_{-2} &= \frac{1}{2}\tau^2
\end{aligned}$$

This concludes our filter design of a C^1 2EF first derivative filter.

In the Appendix we list all the interpolation and first derivative filters that we constructed using different accuracy and smoothness criteria. Because of space constraint we have only given the polynomial coefficients in a matrix M . The filter weights are computed by:

$$\begin{bmatrix} w_{-3}(\tau) \\ w_{-2}(\tau) \\ w_{-1}(\tau) \\ w_0(\tau) \\ w_1(\tau) \\ w_2(\tau) \end{bmatrix} = M \begin{bmatrix} \tau^3 \\ \tau^2 \\ \tau \\ 1 \end{bmatrix}$$

for a cubic filter with 6 weights. The size of M is adapted by the size of the filter and the degree of the polynomial. All filters are laid out in a table where the rows represent the smoothness criteria and the columns represent the accuracy criteria. We have looked at 1EF through 4EF filters and C^0 through C^3 smoothness criteria. We also included filters that were constructed without Condition 3 - smoothness. That simply leads to discontinuous filters. Those filters might be of interest for applications that care about accuracy only, for example in cases where the resulting function is used for measurement, rather than visual inspection. The advantage of these filters is that they are sometimes faster while having only lit-

tle (numerical) error. We have argued in [17] that these filters might even lead to reasonable images under certain conditions and therefore represent an efficient alternative that should not be discarded.

The most general usable filters are probably C^0 , C^1 and 2EF, 3EF filters. For the interpolation filters we find that the most efficient C^1 -3EF filter is the well known Catmull-Rom spline, also found by others to be the most accurate BC-spline. It is also noteworthy that this filter is not the best filter in the class of cubic BC-splines in terms of smoothness. We also found that the BC-filter for which $B=1$ and $C=0$ is a C^2 -2EF filter. Therefore this filter might be preferable over the Catmull-Rom spline for some applications. In order to improve on the Catmull-Rom filter in terms of accuracy one requires 6 filter weights. In order to improve smoothness of the reconstructed function while maintaining the same accuracy, one has to choose at least a fourth degree polynomial. The best filter with just 2 filter weights would be either a 2EF or a C^3 continuous filter.

For derivative filters, the filter C^1 -2EF is probably a good first derivative filter. It is one of the best possible that only requires 4 filter weights and is still only a quadratic filter. In order to improve on it, we would either have to go to 6 filter weights or to a fourth degree polynomial. It is also interesting to note, that this filter is the analytic derivative of the C^2 -2EF interpolation filter, which was a BC-spline with $B=1$ and $C=0$.

5. EXPERIMENTS

The images were rendered employing a simple ray-caster to find the iso-surfaces. The volumes were sampled at an interval of 0.05 voxel lengths. At each sampling point, the ray-caster first applied an interpolation kernel (we used the Catmull-Rom cubic spline) to reconstruct the function at that point. If the reconstructed value was above a pre-set iso-value, the derivative filter was used to compute the 3D gradient. Shading was then performed using the traditional Phong lighting model [8] with diffuse and specular reflections. The obtained color and opacity were composited with the previous ray values, and the ray was terminated after the opacity reached a value close to 1.0. Since both the interpolation and the derivative kernel were separable, for all our filters, the filtering operations could be efficiently performed using a scheme similar to the one given in [1] and [18].

For our experiments we used an analytic data set and an MRI data set. The analytic data set is derived from the same function as the one used by Marschner and Lobb [15]. Since, due to spatial constraints, it is not possible to include the entire set of images that can be obtained using all given filters, summarized in Table 1 and Table 2, we have chosen the discontinuous and C^0 1EF filters as well as the discontinuous and C^0 3EF filters. Fig. 2 (Fig. 2,4,5 in color plates) shows the synthetic data set. In order to better visualize the influence of the filters we also computed the angular error images. For each reconstructed normal we computed the actual normal and recorded their angular difference. The grey value of 255 was displayed for an angular error of 15 degrees. The discontinuous 1EF filter is simply the well known central difference filter, and the discontinuous 3EF filter is the filter that we have found to be a filter yielding better accuracy in our previous work [17]. Here it is clearly visible, that filter design solely based on accuracy criteria will not lead to acceptable images. Adding a simple smoothness

constraint, which is reflected in the C^0 filters, results in very accurate images that are free of visible artifacts. It also becomes very clear, especially in the error images of Fig. 3, that the 3EF filter will lead superior images in terms of numerical accuracy.

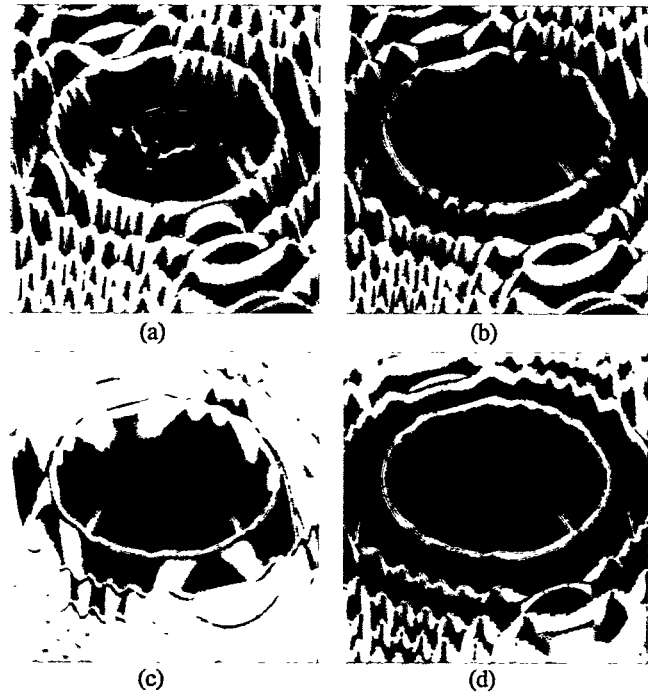


FIGURE 3. Error images of the Marschner Lobb data set rendered using the following derivative filter (a) discontinuous 1EF (b) discontinuous 3EF (c) C^0 -1EF (d) C^0 -3EF; darker colors mean lesser error

The same behavior as for the analytic data set can also be observed for the MRI data set in Fig. 4. This data set is a close up view of an MRI of a human brain. Here, we also fixed the interpolation filter to the Catmull-Rom filter and varied the derivative filter in the same way as we did for the Marschner Lobb images.

Another application requiring smooth reconstruction filters is the size preserving pattern mapping of Kurzion et al. [12]. Here, the problem is to continuously map a texture to a parametric surface or implicit surface, including volumetric iso-surfaces, at a constant density. In the past, only manual mappings were able to perform this task, while this paper introduces an automatic method. The authors use the curvature of a surface at a point in order to continuously vary the scale of the mapped image. This curvature is approximated using the derivative of the underlying function. A C^1 continuous filter is essential for the success of this method as it ensures continuous mapping of texture on the surface. We used a 16^3 grid (a shrunk down version of the original 128^3 head) for calculating the curvature. This means that the head is composed of rectangular patches on which the normal derivative is calculated by the same 4^3 grid samples. Fig. 5a uses the central difference filter, which gives a very poor estimation of the curvature, hence the mapping of the density varies sharply between patches. Fig. 5b uses a C^0 -2EF filter that generates a very constant density across the head, but shows discontinuities along the patch lines. The filter we designed for this application is the C^1 continuous 2EF derivative filter of Table 2. Fig. 5c shows an application of this filter and we observe that all previous problems no longer exist.

6. FURTHER RESEARCH

Since, in this paper, we restricted ourselves to piecewise polynomial filters, we would like to explore different basis functions. Especially using trigonometric basis functions like *sine* and *cosine*, we would like to explore possible similarities to windowed *Sinc* and *Cosc* filters. We hope that this might improve the efficiency of the filters, maintaining the same smoothness and accuracy while using fewer filter weights.

Our accuracy criteria, as outlined here, have also been used to specify and design filters in other domains, especially in wavelet-based multiresolution analyses, by Daubechies and others [5]. We hope to explore the effect of our smoothness criteria also for defining new basis functions in the wavelet domain and to study its effects on the wavelet transform.

In applications of computer graphics, especially scientific visualization, the final image is not just influenced by the interpolation and derivative method, but sometimes also by a shading equation and compositing operations. Therefore, it is necessary to analyze the overall error expressed in the L^2 error norm. We are working on developing better tools to study this error. Once we are able to characterize the complete rendering pipeline, we might be able to adapt the reconstruction filters in a way to compensate for the overall error and to produce better images.

7. CONCLUSIONS

In this work, we have introduced design criteria for the design of optimal and smooth reconstruction filters. It was demonstrated, contrary to previous beliefs, that it is not sufficient to base filter designs just on accuracy criteria. Rather, our design criteria are not only the accuracy of the reconstructed function, but also its characterization as a function in the space C^M of all M -times continuous derivable functions. We further demonstrated, by ways of an example, a C^1 2EF filter, how such a filter can be constructed. We then provided a table, listing all optimal interpolation and derivative filters that match smoothness and accuracy criteria of up to C^3 continuous and 4EF, respectively. These tables may serve as a guide on what filters should be used for a certain application. The filters listed in these tables go much beyond the very popular class of piecewise polynomial filters, described in the past by Mitchell and Netravali's BC-splines [16]. Furthermore, this paper gives the practitioner an easy way to design specific filters to match their specific application, by determining a very application oriented set of criteria - smoothness and accuracy.

8. ACKNOWLEDGMENTS

We thank Prof. Ed Overman of the Department of Mathematics at Ohio State University for providing useful background information and important references. We thank Prof. Wayne Carlson and the Advanced Computing Center for the Arts and Design for the use of their computing facilities and Prof. Robert Moorehead of the NSF Engineering Research Center, Mississippi State University for providing encouragement and support.

9. REFERENCES

- [1] Bentum M.J., Lichtenbelt B.B.A., Malzbender T., "Frequency Analysis of Gradient Estimators in Volume Rendering", *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242-254, September 1996.
- [2] Bracewell R.N., *Two Dimensional Imaging*, Prentice Hall Inc., Englewoods Cliffs, NJ, 1995.
- [3] Bronstein E.N., Semendjajew K.A., "Taschenbuch der Mathematik," 22nd ed., G. Grosche, V. Ziegler, B. Ziegler: editors, BSB B.G. Teubner Verlagsgesellschaft, Leipzig, 1985.
- [4] Carlbom I., "Optimal Filter Design for Volume Reconstruction and Visualization", *Proceedings of IEEE Conference on Visualization '93*, pp. 54-61, October 1993.
- [5] Daubechies I., *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, PA, 1992.
- [6] Drebin R.A., L. Carpenter L., Hanrahan P., "Volume Rendering", *Computer Graphics*, 22(4):51-58, August 1988.
- [7] Dutta Roy S.C., Kumar B., "Digital Differentiators", in *Handbook of Statistics*, N. K. Bise and C. R. Rao eds., vol. 10: 159-205, 1993.
- [8] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., *Computer Graphics, Principles and Practice*, Second Edition, Addison-Wesley, Reading, Massachusetts, 1990.
- [9] Goss M.E., "An Adjustable Gradient Filter for Volume Visualization Image Enhancement", *Proceedings of Graphics Interface '94*, pp. 67-74, Toronto, Ontario, 1994.
- [10] Hamming R.W., *Digital Filters*, Prentice Hall Inc., Second Edition, Englewoods Cliffs, NJ, 1983.
- [11] Keys R.G., "Cubic Convolution Interpolation for Digital Image Processing", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29(6):1153-1160, December 1981.
- [12] Kurzion Y., Möller T., Yagel R., "Size Preserving Pattern Mapping", in *IEEE Conference on Visualization '98*.
- [13] Levoy M., "Display of Surfaces from Volume Data", *IEEE Computer Graphics and Applications*, 8(5):29-37, May 1988.
- [14] Machiraju R.K., Yagel R., "Reconstruction Error Characterization and Control: A Sampling Theory Approach", *IEEE Transactions on Visualization and Computer Graphics*, TVCG 2(4):364-376, December 1996.
- [15] Marschner S.R. and Lobb R.J., "An Evaluation of Reconstruction Filters for Volume Rendering", *Proceedings of Visualization '94*, IEEE CS Press, pp. 100-107, October 1994.
- [16] Mitchell D.P., Netravali A.N., "Reconstruction Filters in Computer Graphics", *Computer Graphics*, 22(4):221-228, August 1988.
- [17] Möller T., Machiraju R.K., Mueller K., Yagel R., "Evaluation and Design of Filters Using a Taylor Series Expansion", *IEEE Transactions on Visualization and Computer Graphics*, ITVCG 3(2): 184-199, June 1997.
- [18] Möller T., Machiraju R.K., Mueller K., Yagel R., "A Comparison Of Normal Estimation Schemes", *Proceedings of IEEE Conference on Visualization '97*, pp. 19-26, October 1997.
- [19] Oppenheim A.V., Schaffer R.W., *Discrete-Time Signal Processing*, Prentice Hall Inc., Englewoods Cliffs, NJ, 1989.
- [20] Park S.K., Schowengerdt R.A., "Image Reconstruction by Parametric Cubic Convolution", *Computer Vision, Graphics, and Image Processing*, vol. 23:258-272, 1983.
- [21] Warsi Z.U.A., *Fluid Dynamics: Theoretical and Computational Approaches*, CRC Press, Boca Raton, FL, 1992.

10. APPENDIX

10.1 Interpolation filters

TABLE 1. the new actual filters for interpolation filters

	1EF	2EF	3EF	4EF
C-	$\begin{bmatrix} w_{-1} \\ w_0 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{4} & 2a - \frac{1}{4} & -a \\ \frac{1}{4} & -(6a - \frac{5}{4}) & 3a \\ \frac{1}{4} & 6a - \frac{3}{4} & -(3a - 1) \\ \frac{1}{4} & -(2a + \frac{1}{4}) & a \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & 0 & \frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -1 & \frac{1}{2} & 1 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 \end{bmatrix}$
C0	$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{5}{4} & 0 \\ \frac{1}{4} & \frac{3}{4} & 1 \\ \frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & 0 & \frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -1 & \frac{1}{2} & 1 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} & 0 \end{bmatrix}$
C1	$\begin{bmatrix} -2 & 3 & 0 & 0 \\ 2 & -3 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{3}{2} & 2 & \frac{1}{2} & 0 \\ \frac{3}{2} & \frac{5}{2} & 0 & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} -a & (\frac{3}{2}a - \frac{1}{24}) & 0 & 0 \\ (5a + \frac{1}{6}) & -(\frac{15}{2}a - \frac{1}{8}) & \frac{1}{12} & \frac{1}{2}a - \frac{1}{24} \\ -(10a + \frac{1}{2}) & 15a + \frac{5}{12} & \frac{2}{3} & -(2a - \frac{1}{6}) \\ 10a + \frac{1}{2} & -(15a + \frac{13}{12}) & 0 & 3a + \frac{3}{4} \\ -(5a + \frac{1}{6}) & \frac{15}{2}a + \frac{5}{8} & \frac{2}{3} & -(2a - \frac{1}{6}) \\ a & -(\frac{3}{2}a + \frac{1}{24}) & \frac{1}{12} & \frac{1}{2}a - \frac{1}{24} \end{bmatrix}$
C2	$\begin{bmatrix} 6 & -15 & 10 & 0 & 0 & 0 \\ -6 & 15 & -10 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & -1 & 0 & \frac{2}{3} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & \frac{1}{6} \end{bmatrix}$	$\begin{bmatrix} -1 & \frac{5}{2} & \frac{3}{2} & 0 & 0 & 0 \\ 3 & \frac{15}{2} & \frac{9}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ -3 & \frac{15}{2} & \frac{9}{2} & -1 & 0 & 1 \\ 1 & \frac{5}{2} & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{36} & 0 & 0 & 0 \\ \frac{11}{36} & \frac{1}{12} & \frac{1}{12} & \frac{1}{36} \\ \frac{7}{9} & \frac{5}{6} & \frac{2}{3} & \frac{1}{9} \\ \frac{7}{9} & \frac{3}{2} & 0 & \frac{5}{6} \\ \frac{11}{36} & \frac{5}{6} & \frac{2}{3} & \frac{1}{9} \\ \frac{1}{36} & \frac{1}{12} & \frac{1}{12} & \frac{1}{36} \end{bmatrix}$
C3	$\begin{bmatrix} -20 & 70 & -84 & 35 & 0 & 0 & 0 & 0 \\ 20 & -70 & 84 & -35 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{10} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{3}{10} & \frac{3}{4} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{20} \\ \frac{3}{10} & \frac{3}{4} & 0 & -1 & 0 & \frac{7}{10} \\ \frac{1}{10} & \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{3}{20} \end{bmatrix}$	$\begin{bmatrix} 3 & \frac{21}{2} & \frac{25}{2} & -5 & 0 & 0 & 0 & 0 \\ -9 & \frac{63}{2} & \frac{75}{2} & 15 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 9 & \frac{63}{2} & \frac{75}{2} & -15 & 0 & -1 & 0 & 1 \\ -3 & \frac{21}{2} & \frac{25}{2} & 5 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{30} & \frac{1}{16} & 0 & 0 & 0 & 0 \\ \frac{1}{6} & \frac{17}{48} & \frac{1}{12} & \frac{1}{24} & \frac{1}{12} & \frac{7}{240} \\ \frac{1}{3} & \frac{19}{24} & \frac{1}{6} & \frac{2}{3} & \frac{2}{3} & \frac{7}{60} \\ \frac{1}{3} & \frac{7}{8} & 0 & \frac{5}{4} & 0 & \frac{33}{40} \\ \frac{1}{6} & \frac{23}{48} & \frac{1}{6} & \frac{2}{3} & \frac{2}{3} & \frac{7}{60} \\ \frac{1}{30} & \frac{5}{48} & \frac{1}{12} & \frac{1}{24} & \frac{1}{12} & \frac{7}{240} \end{bmatrix}$

10.2 First Derivative filters

TABLE 2. the new actual derivative filters

	1EF	2EF	3EF	4EF
C-	$\begin{bmatrix} w_{-2} \\ w_{-1} \\ w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & a \\ \frac{1}{2} & \frac{1}{2} - 3a \\ \frac{1}{2} & 3a \\ \frac{1}{2} & -\frac{1}{2} - a \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{6} \\ \frac{3}{2} & 1 & 1 \\ \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12} & a & -(a+\frac{1}{6}) & b \\ \frac{1}{4} & -(5a-\frac{1}{4}) & 5a+\frac{3}{4} & -(5b+\frac{1}{12}) \\ \frac{1}{6} & 10a-\frac{1}{2} & -(10a+\frac{1}{3}) & 10b+\frac{2}{3} \\ \frac{1}{6} & -10a & 10a-\frac{5}{6} & -10b \\ \frac{1}{4} & 5a+\frac{1}{2} & -(5a-\frac{1}{2}) & 5b-\frac{2}{3} \\ \frac{1}{12} & -(a+\frac{1}{4}) & a+\frac{1}{12} & -(b-\frac{1}{12}) \end{bmatrix}$
C0	$\begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} a & -(a+\frac{1}{12}) & 0 \\ -(5a-\frac{1}{2}) & 5a+\frac{1}{4} & -\frac{1}{12} \\ 10a-\frac{3}{2} & -(10a-\frac{5}{6}) & \frac{2}{3} \\ -(10a-\frac{3}{2}) & 10a-\frac{13}{6} & 0 \\ 5a-\frac{1}{2} & -(5a-\frac{5}{4}) & -\frac{2}{3} \\ -a & a-\frac{1}{12} & \frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12} & a & -(a+\frac{1}{6}) & 0 \\ \frac{1}{4} & -(5a-\frac{1}{4}) & 5a+\frac{3}{4} & -\frac{1}{12} \\ \frac{1}{6} & 10a-\frac{1}{2} & -(10a+\frac{1}{3}) & \frac{2}{3} \\ \frac{1}{6} & -10a & 10a-\frac{5}{6} & 0 \\ \frac{1}{4} & 5a+\frac{1}{2} & -(5a-\frac{1}{2}) & -\frac{2}{3} \\ \frac{1}{12} & -(a+\frac{1}{4}) & a+\frac{1}{12} & \frac{1}{12} \end{bmatrix}$
C1	$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{3}{2} & -2 & 0 \\ \frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{3}{2} & -2 & 0 \\ \frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12} & 0 & 0 \\ \frac{11}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{7}{3} & \frac{5}{3} & \frac{2}{3} \\ \frac{7}{3} & -3 & 0 \\ \frac{11}{12} & \frac{5}{3} & \frac{2}{3} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{12} & \frac{1}{6} & 0 & 0 \\ \frac{1}{4} & \frac{13}{12} & \frac{1}{12} & \frac{1}{12} \\ \frac{1}{6} & \frac{13}{6} & \frac{4}{3} & \frac{2}{3} \\ \frac{1}{6} & \frac{5}{3} & -\frac{5}{2} & 0 \\ \frac{1}{4} & \frac{1}{3} & \frac{4}{3} & -\frac{2}{3} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{bmatrix}$
C2	$\begin{bmatrix} \frac{1}{2} & 1 & 0 & 0 & 0 \\ \frac{3}{2} & -3 & 0 & 1 & \frac{1}{2} \\ \frac{3}{2} & 3 & 0 & -2 & 0 \\ \frac{1}{2} & -1 & 0 & 1 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 1 & 0 & 0 & 0 \\ \frac{3}{2} & -3 & 0 & 1 & \frac{1}{2} \\ \frac{3}{2} & 3 & 0 & -2 & 0 \\ \frac{1}{2} & -1 & 0 & 1 & -\frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{5}{6} & \frac{17}{12} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \\ \frac{5}{3} & \frac{19}{6} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{5}{3} & \frac{7}{2} & 0 & \frac{5}{2} & 0 \\ \frac{5}{6} & \frac{23}{12} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{1}{6} & \frac{5}{12} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{6} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{5}{6} & \frac{17}{12} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \\ \frac{5}{3} & \frac{19}{6} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{5}{3} & \frac{7}{2} & 0 & \frac{5}{2} & 0 \\ \frac{5}{6} & \frac{23}{12} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{1}{6} & \frac{5}{12} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \end{bmatrix}$
C3	$\begin{bmatrix} \frac{3}{4} & \frac{5}{4} & 0 & 0 & 0 & 0 \\ \frac{3}{4} & 0 & \frac{5}{2} & \frac{5}{4} & \frac{1}{2} \\ \frac{3}{4} & \frac{15}{4} & 5 & 0 & \frac{5}{2} & 0 \\ \frac{3}{4} & \frac{5}{2} & \frac{5}{2} & \frac{5}{4} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & -3 & \frac{5}{2} & 0 & 0 & 0 \\ -3 & 9 & \frac{15}{2} & 0 & 0 & \frac{1}{2} \\ 3 & -9 & \frac{15}{2} & 0 & 0 & -2 \\ -1 & 3 & -\frac{5}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{3}{16} & \frac{13}{48} & 0 & 0 & 0 & 0 \\ \frac{9}{16} & \frac{5}{12} & \frac{19}{24} & \frac{1}{4} & \frac{7}{48} & \frac{1}{12} \\ \frac{3}{8} & \frac{25}{24} & \frac{19}{6} & \frac{1}{2} & \frac{19}{12} & \frac{2}{3} \\ \frac{3}{8} & \frac{35}{12} & \frac{19}{4} & 0 & \frac{23}{8} & 0 \\ \frac{9}{16} & \frac{115}{48} & \frac{19}{6} & \frac{1}{2} & \frac{19}{12} & \frac{2}{3} \\ \frac{3}{16} & \frac{2}{3} & \frac{19}{24} & \frac{1}{4} & \frac{7}{48} & \frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{4} & \frac{3}{4} & \frac{7}{12} & 0 & 0 & 0 & 0 \\ \frac{5}{4} & \frac{15}{4} & \frac{35}{12} & \frac{1}{6} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \\ \frac{5}{2} & \frac{15}{2} & \frac{35}{6} & \frac{2}{3} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{5}{2} & \frac{15}{2} & \frac{35}{6} & 1 & 0 & \frac{5}{2} & 0 \\ \frac{5}{4} & \frac{15}{4} & \frac{35}{12} & \frac{2}{3} & \frac{1}{2} & \frac{4}{3} & \frac{2}{3} \\ \frac{1}{4} & \frac{3}{4} & \frac{7}{12} & \frac{1}{6} & \frac{1}{4} & \frac{1}{12} & \frac{1}{12} \end{bmatrix}$

Author Index

(Color plate pages are shown in *italics*.)

Bajaj, C.95, 172
 Banks, David C.103, 173
 Behrens, Uwe39, 165
 Bitter, Ingmar55, 167
 Breen, David E.7, 161
 Carvajal, Marcelo127, 176
 Chen, Baoquan55, 167
 Cohen-Or, Daniel15, 162
 Crawfis, Roger63, 168
 Dachtelle, Frank55, 167
 Durkin, James W.79, 170
 Filippov, Vassily119, 175
 Gagvani, Nikhil47, 166
 Gibson, Sarah F. F.23, 163
 Goss, Michael E.135, 177
 Huang, Jian63, 119, 168, 175
 Kaufman, Arie55, 111, 167, 174
 Kenchammana-Hosekote, Dilip47, 166
 Kindlmann, Gordon79, 170
 Kreeger, Kevin55, 167
 Kurzion, Yair119, 143, 175, 178
 Lauer, Hugh C.31, 164
 Ma, Feng71, 169
 Machiraju, Raghu143, 178
 Malzbender, Thomas135, 177
 Mauch, Sean7, 161

Mitchell, Joseph S. B.87, 171
 Mueller, Klaus143, 178
 Möller, Torsten143, 178
 Ogata, Masato31, 164
 Ohkami, TakaHide31, 164
 Pascucci, V.95, 172
 Pfister, Hanspeter31, 164
 Rabbiolo, G.95, 172
 Ratering, Ralf39, 165
 Schikore, D.95, 172
 Shibolet, Omer15, 162
 Silva, Cláudio T.87, 171
 Silver, Deborah47, 166
 Srámek, Milos111, 174
 Stredney, Don63, 168
 Sánchez, Ricardo127, 176
 Tang, Zesheng71, 169
 Tsang, Wai Wan71, 169
 Wang, Wenping71, 169
 Weigle, Chris103, 173
 Whitaker, Ross T.7, 161
 Williams, Peter L.87, 171
 Wittenbrink, Craig M.135, 177
 Xia, Shaowei71, 169
 Yagel, Roni119, 143, 175, 178

Cover Image Credits

Front Cover

Virtual Endoscopy - A voxel-based colon, colored using a texture of a human tissue.

Reference: "Coloring Voxel-Based Objects for Virtual Endoscopy," Omer Shibolet, Daniel Cohen-Or, pp. 15-22.

Back Cover

Top:

Volume Rendering of a Bolt Using Wavelet Based Adaptive Interpolation

Reference: "Wavelet Based Adaptive Interpolation For Volume Rendering," Ricardo Sánchez, Marcelo Carvajal, pp. 127-134.

Middle Left:

Rendered images of the sloth CT volume with SOM-PNN segmentation.

Reference: "Probabilistic Segmentation of Volume Data for Visualization Using SOM-PNN Classifier," Feng Ma, Wenping Wang, Wai Wan Tsang, Zesheng Tang, Shaowei Xia, pp. 71-78.

Middle Right:

The hypervolume rendering technique combined with the scalar topology enhancement is used to produce four different views of a 5-dimensional scalar field. The dataset is the interaction potential of a receptor protein with respect to a ligand molecule for any independent protein translation along the X,Y,Z axes and rotation around the X and Y axes (assuming rough symmetry of the ligand with respect to the Z axis). The transfer function of the hypervolume is modulated to enhance in red, regions of attraction of the ligand towards the receptor, and in blue, regions of high repulsion (interior of the receptor). Green regions show configurations of free movement. The scalar topology enhances the existence of short steepest descent paths from high attraction to high repulsion regions that might be favorable for the docking of the two molecules.

Reference: "Hypervolume Visualization: a Challenge in Simplicity," C. Bajaj, V. Pascucci, G. Rabbio, D. Schikore, pp. 95-102.

Bottom:

Left: Separate interpolation of color and opacity. Middle: Opacity-weighted interpolation of colors. Right: Normalized difference image. Data courtesy of Dr. Ramani Pichumani, Stanford University.

Reference: "Opacity-Weighted Color Interpolation for Volume Sampling," Craig M. Wittenbrink, Thomas Malzbender, Michael E. Goss, pp. 135-142.

Color Plate Section



Figure 6: CSG model from the original Constructive Cubes paper.

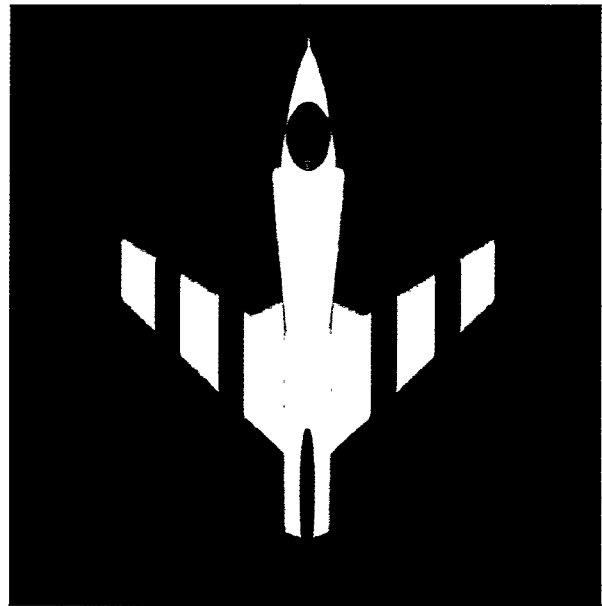


Figure 8: X-29 CSG model surface evaluation utilizing distance volumes.

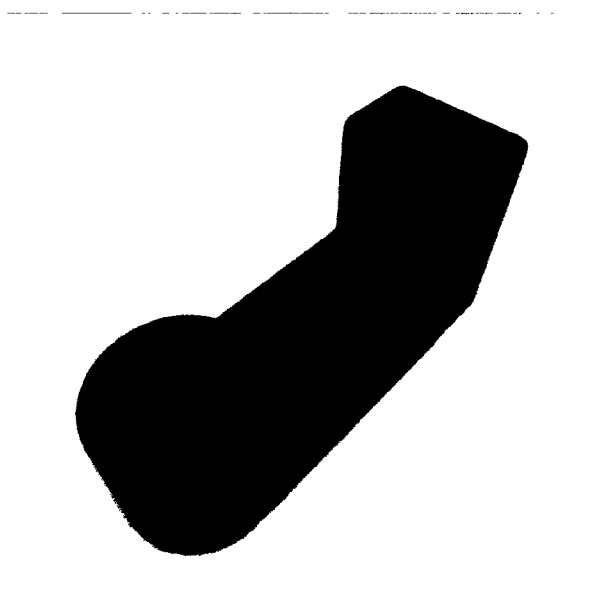


Figure 7: Improved surface evaluation utilizing distance volumes on a similar model.

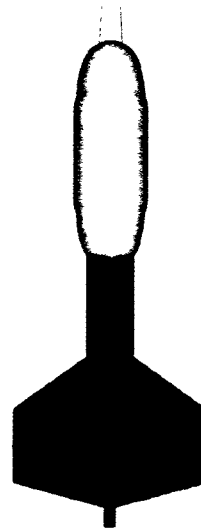
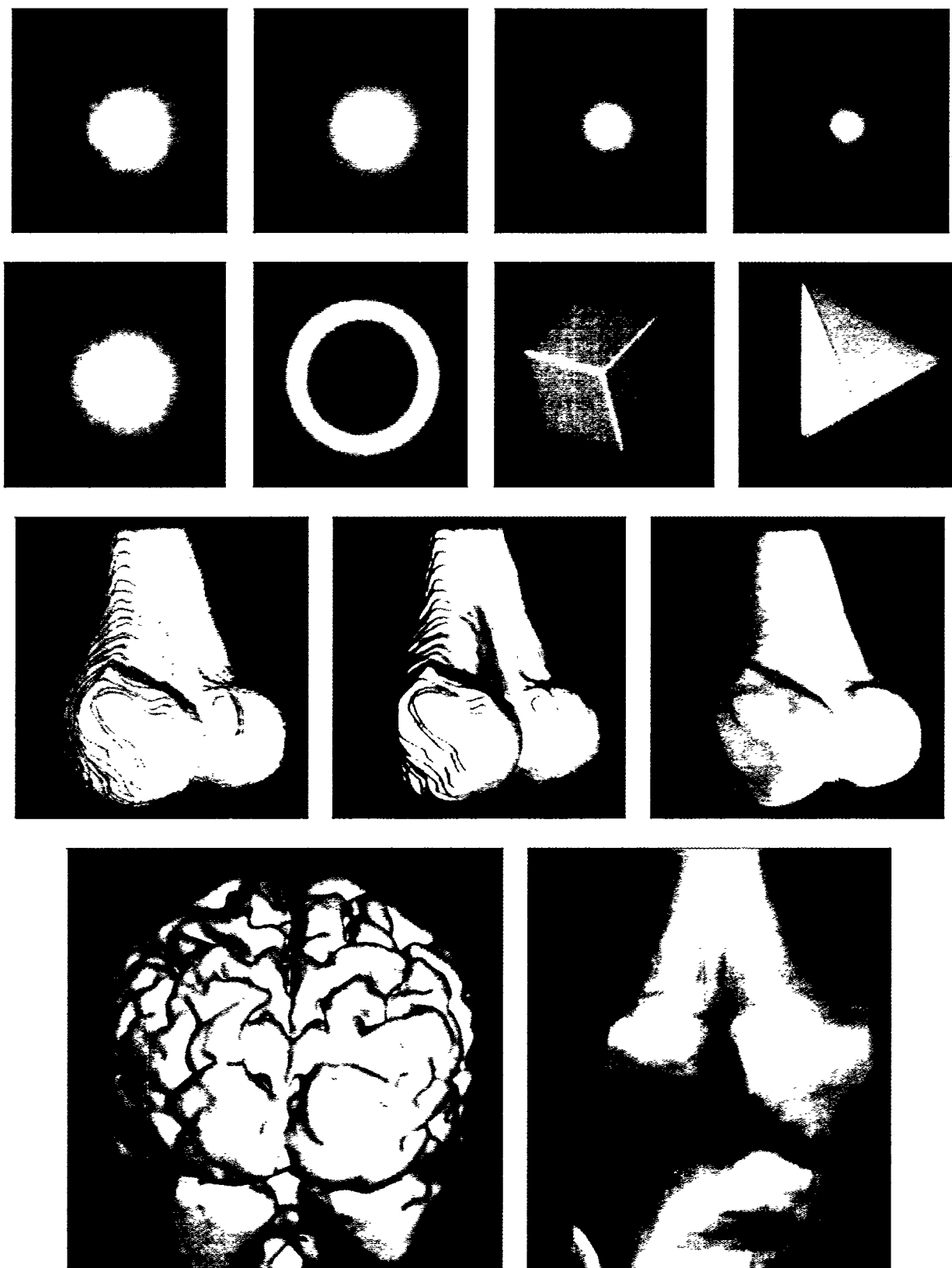


Figure 9: Dart CSG model surface evaluation utilizing distance volumes.

3D Scan Conversion of CSG Models into Distance Volume
David E. Breen, Sean Mauch, Ross T. Whitaker



Coloring Voxel-Based Objects for Virtual Endoscopy
Omer Shibolet, Daniel Cohen-Or



Using Distance Maps for Accurate Surface Reconstruction in Sampled Volumes
Sarah F. F. Gibson

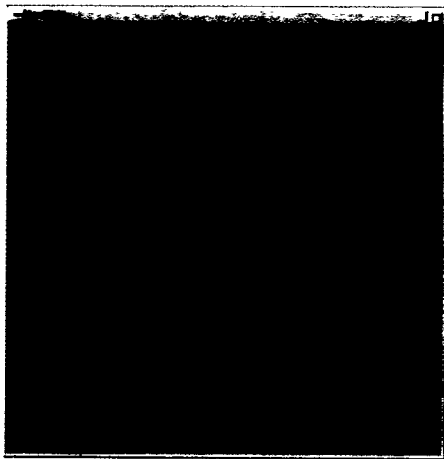


Figure 14: Perspective projection with $2 \times 2 \times 2$ convolution.

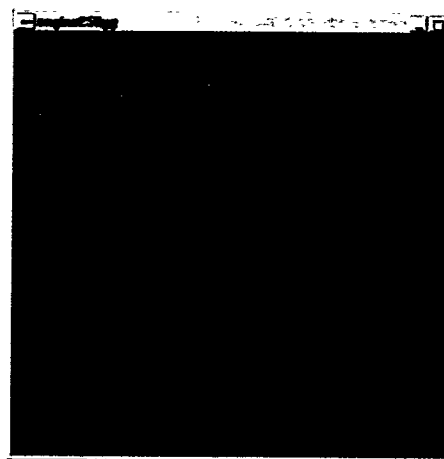


Figure 17: Perspective projection with $3 \times 3 \times 3$ convolution.

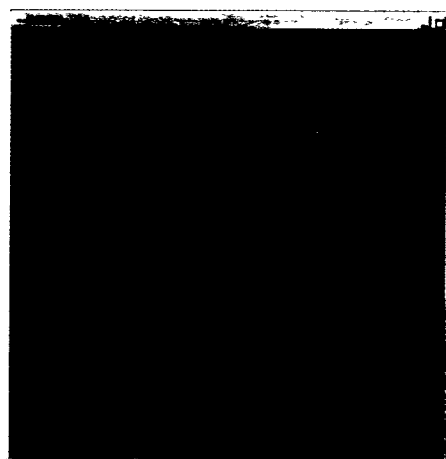


Figure 15: Perspective projection by resampling the nearest neighbor voxels.

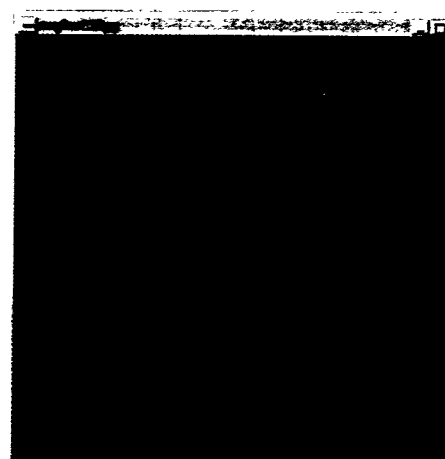


Figure 18: Parallel projection with $3 \times 3 \times 3$ convolution.

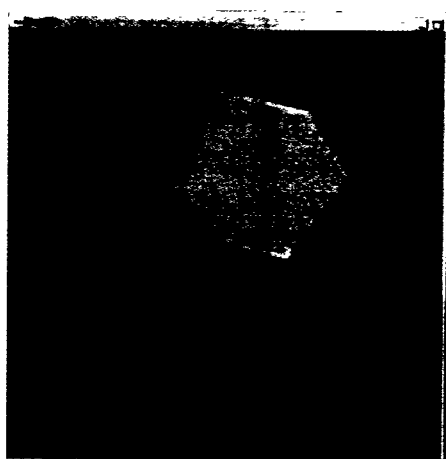


Figure 16: Perspective projection with $3 \times 3 \times 3$ convolution.

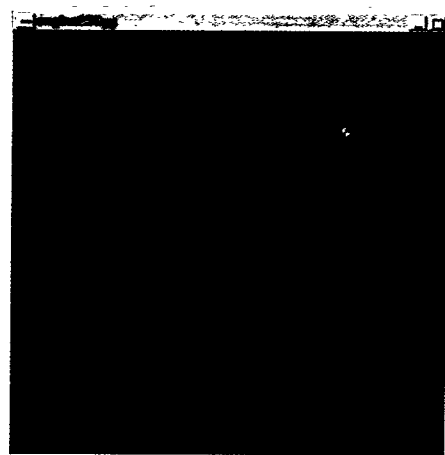


Figure 19: Screen-to-Object perspective projection by interpolations using multi-resolution datasets.

A Real-Time Volume Rendering Architecture Using an Adaptive Resampling Scheme
for Parallel and Perspective Projections

Masato Ogata, TakaHide Ohkami, Hugh C. Lauer, Hanspeter Pfister

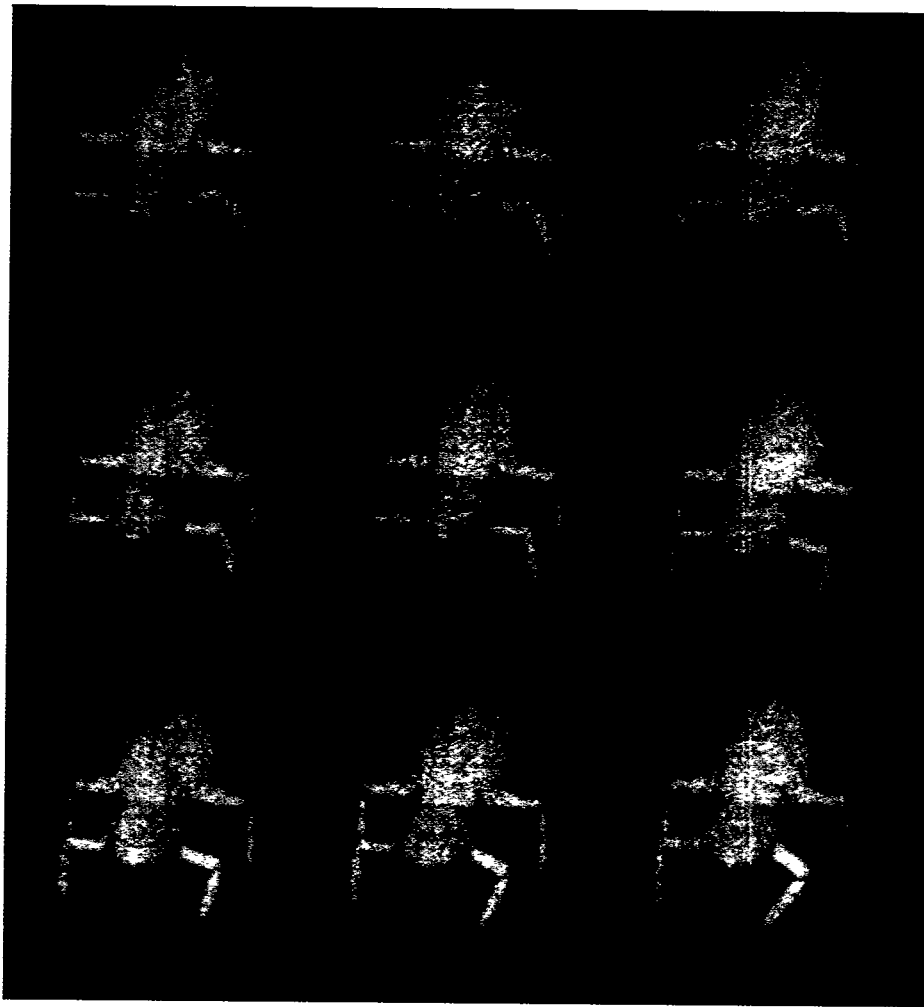


Figure 6: Animating the limb of a volumetric insect - Note how the left front limb folds into the body.

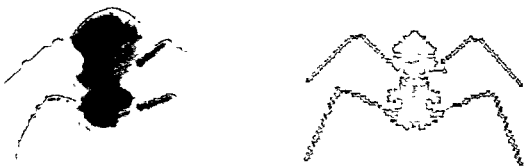


Figure 7: Volumetric Insect and its Skeleton Tree.

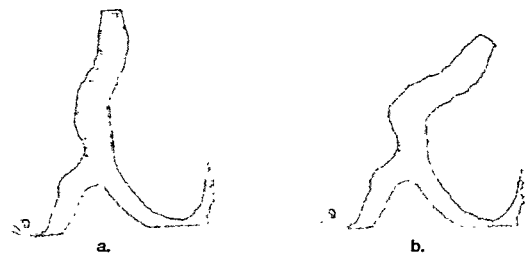
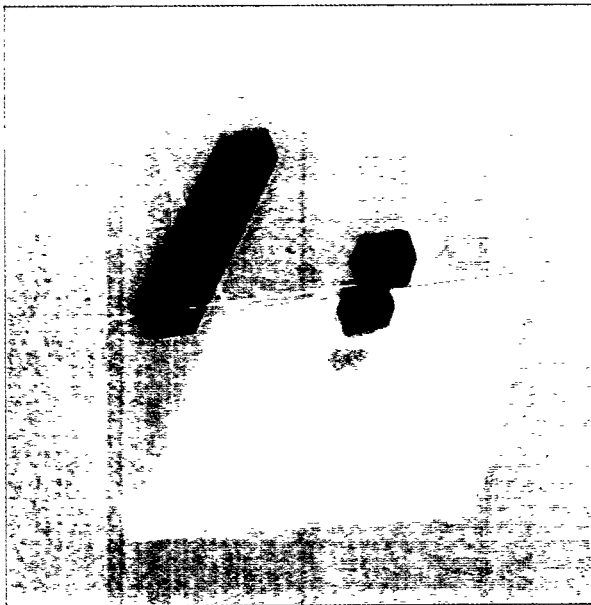
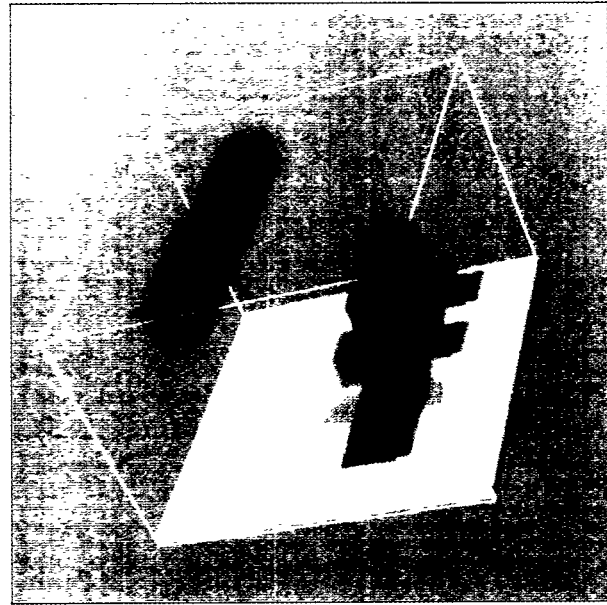


Figure 8: Deforming a Medical Dataset: (a) Part of the Human Trachea (512x512x181, 119,934 voxels) (b) with a sharp bend. The bend is smooth despite the sharp angle.

Volume Animation using the Skeleton Tree
Nikhil Gagvani, Dilip Kenchammana-Hosekote, Deborah Silver

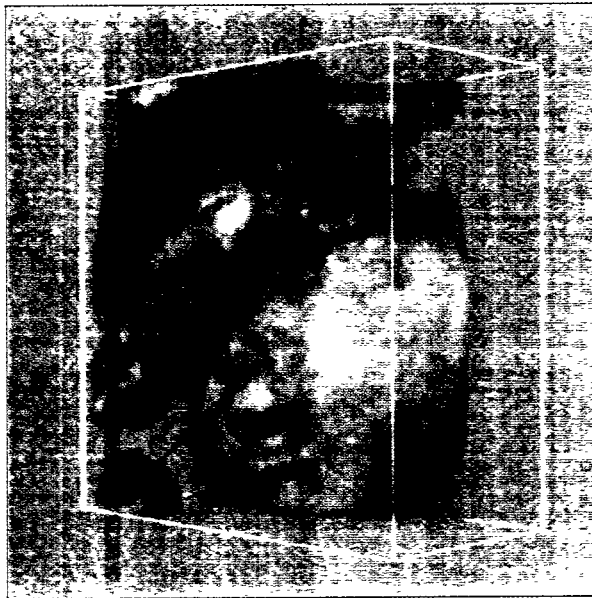


(a) Data-set of a voxelized geometry without shadows

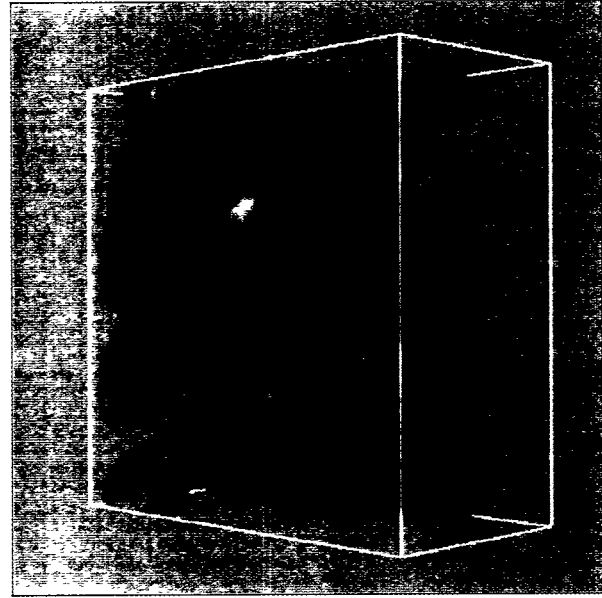


(b) Data-set of a voxelized geometry with shadows

Figure 7: Comparison of shadowed and unshadowed texture-based volume rendering of a geometrical data-set. This $128 \times 128 \times 64$ voxel sized data-set was generated to control the correct calculation of the shadows. The images were rendered using 256 parallel textured planes. In the scene there is the large red block on the top with the four smaller green blocks below, that have different transparencies. In the shadowed image you can how the different green blocks cast different shadows on the ground and how the shadow of the red block appears different on the green blocks. You can also see the fake diffuse illumination effect here, because surfaces that face away from the light source are darker than others.



(a) Ultrasound data-set of fetus rendered without shadows



(b) Ultrasound data-set of fetus rendered with shadows

Figure 8: Screenshots from a volume rendering with the $128 \times 128 \times 64$ voxel sized data-set mapped as a 3D-texture onto 256 parallel planes. Note, how the shadow casts from the fetus' arm onto its face. Although the shadowed image looks more realistic, there are some details covered by the shadow. But the data-set can now be explored by interactively moving the light source, so details will be revealed that cannot be seen in the unshadowed image.

Adding Shadows to a Texture-Based Volume Renderer
Uwe Behrens, Ralf Ratering

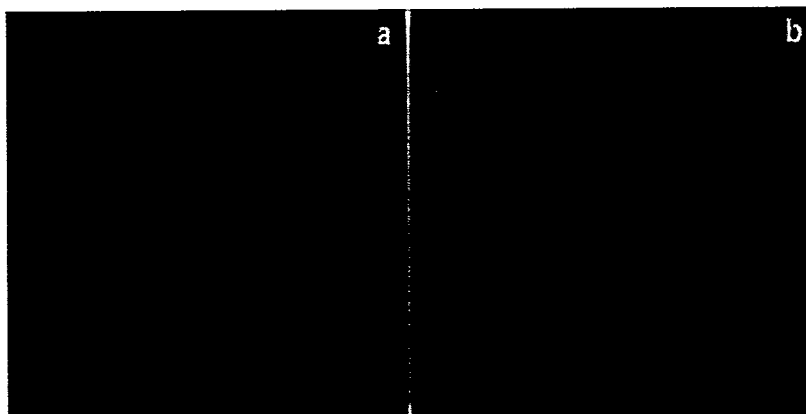


Figure 11: Rendering the test volume with (a) ER-Perspective with Bartlett filter and (b) Novins et al. algorithm with a box filter. There is slight, but noticeable, reduced aliasing for the Bartlett filter.

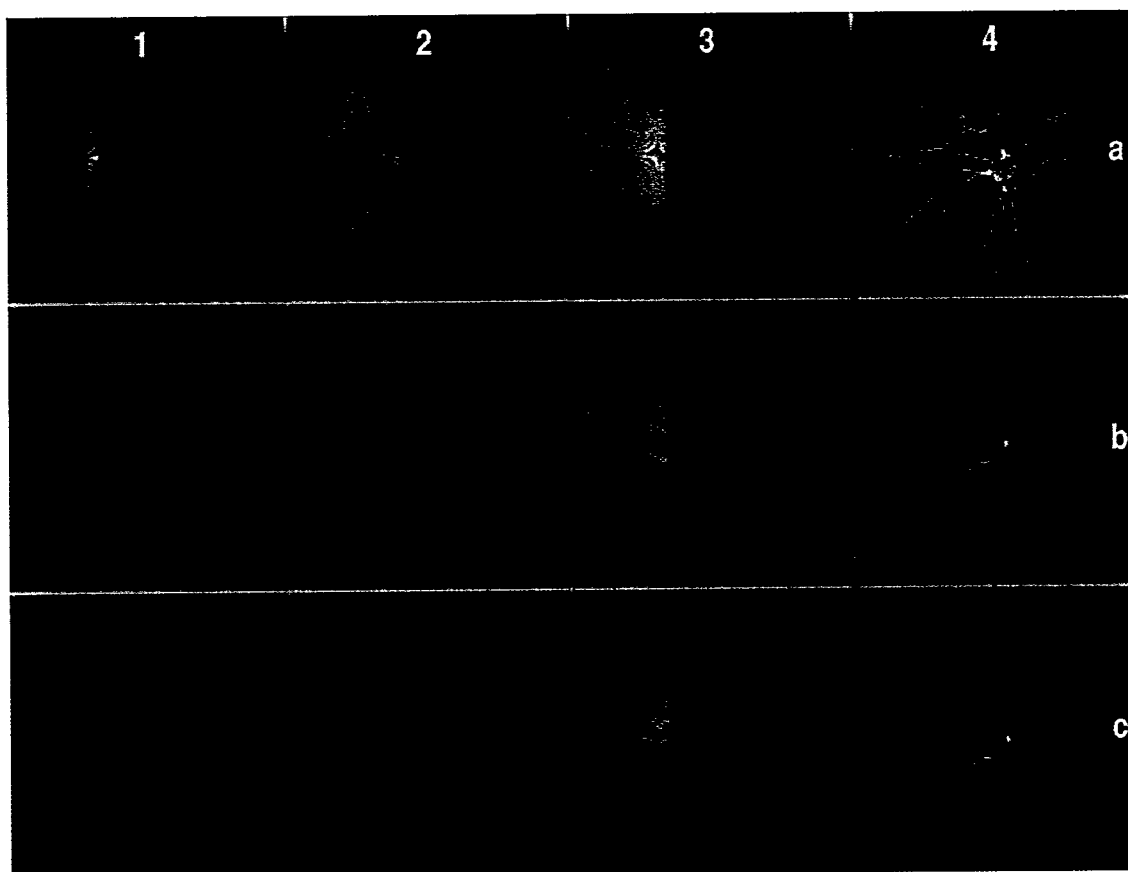


Figure 12: Comparison of perspective volume rendering methods. Rows are different methods: (a) Under-sampling, (b) ER-Perspective, (c) Oversampling. Column 4 is a Lateral Geniculate Nucleus (LGN) neuron. Columns 1 through 3 are our 256^3 test volume with different configuration parameters: (1) A volume with 3^3 checked subblocks, the eyepoint is at $(128, 128, -128)$ rendered with a 90 deg field-of-view. (2) A volume with 5^3 subblocks rendered from $(128, 128, -64)$ with a 127 deg field-of-view. (3) Also a volume with 5^3 subblocks although it is rendered from $(128, 128, -128)$ with a 90 deg field-of-view.

Adaptive Perspective Ray Casting

Kevin Kreeger, Ingmar Bitter, Frank Dachtler, Baoquan Chen, Arie Kaufman

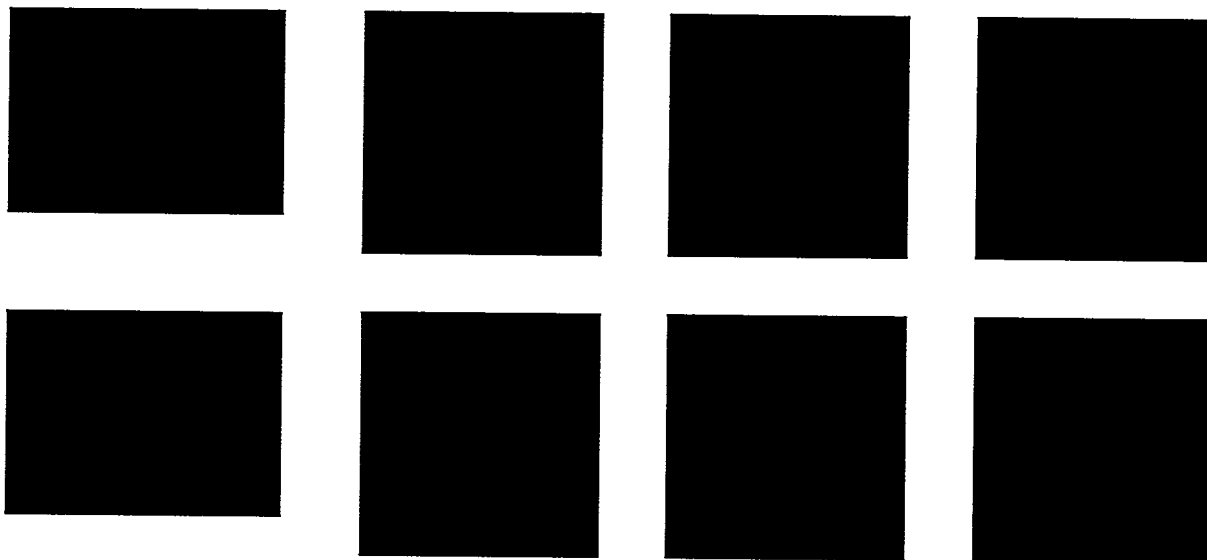


Figure 10c. Images obtained from a virtual navigation inside human nasal cavity. The images in the top row are rendered with conventional splatting algorithm. The images in the bottom row are rendered using our new algorithm for edge preservation.

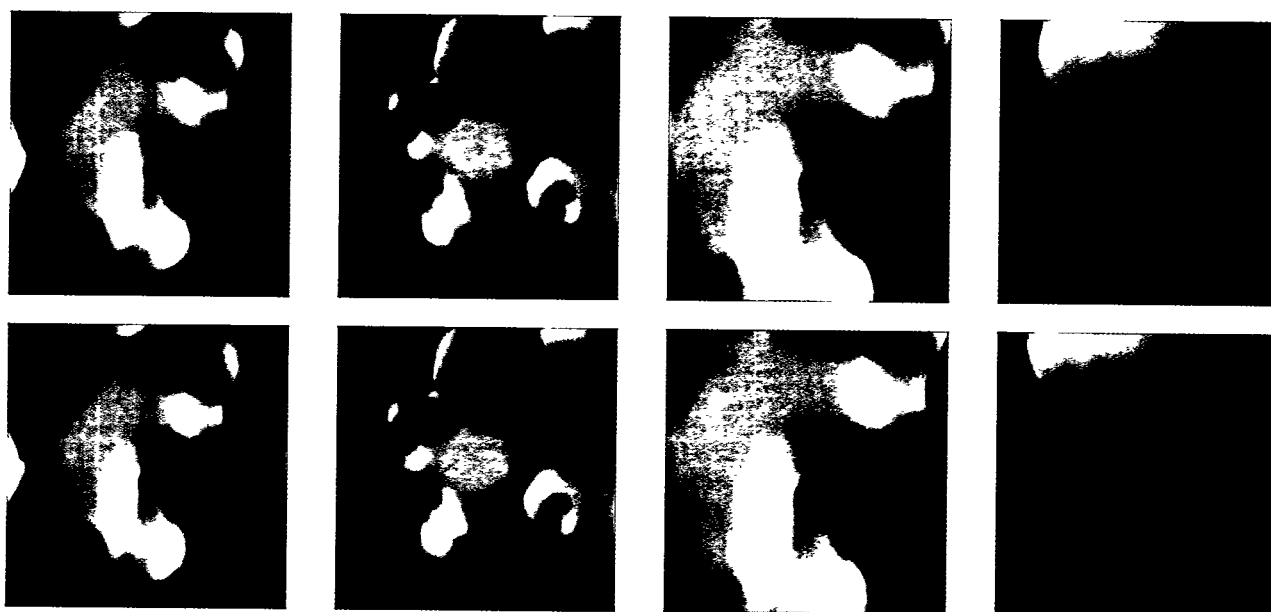


Figure 11c. Images acquired from a navigation through the Hipip dataset ($64 \times 64 \times 64$). We first threshold the dataset and get the iso-contour with values between 0.0017 and 0.5, then render with the splatting approach. The top row corresponds to the conventional splatting, the bottom row images are results of our new algorithm for edge preservation.

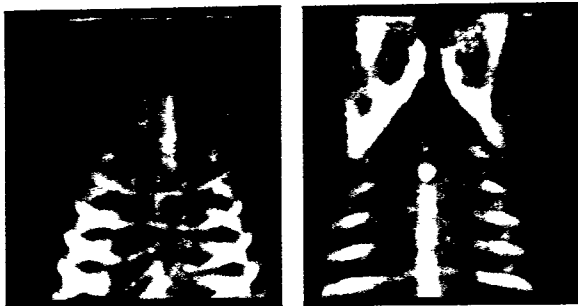


Fig. 3. Rendered images of the sloth CT volume with SOM-PNN segmentation.

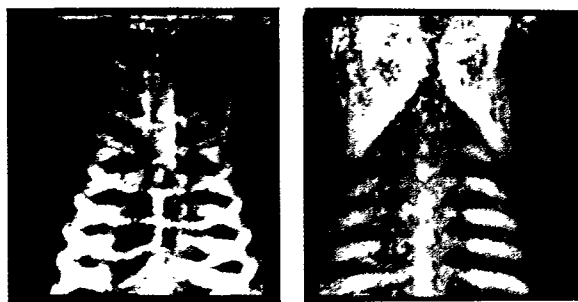


Fig. 4. Rendered images of the sloth CT volume with ML segmentation.

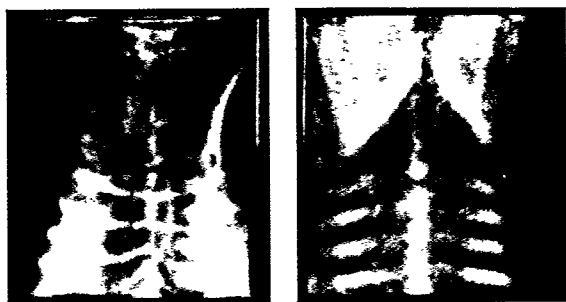


Fig. 5. Rendered images of the sloth CT volume with PNN segmentation.

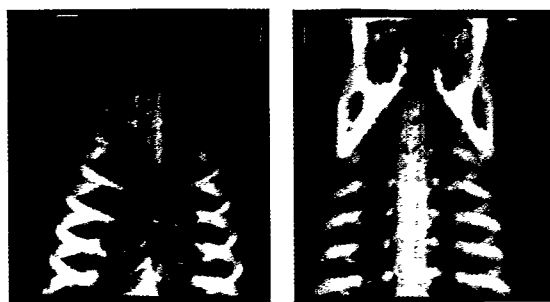


Fig. 6. Rendered images of the sloth CT volume with SOM segmentation.



Fig. 8. Coronally clipped views of the MRI brain case 112_2 with manual segmentation, SOM-PNN, ML, PNN and SOM segmentations respectively.



Figure 12: Analysis of a magnetic resonance image of a portion of the brain. On the left is a scatterplot of data value and gradient magnitude; no clear boundaries are evidenced. In the middle is an automatically generated opacity function of data value, and on the right is the rendered image. The goal of the visualization was to find the aneurysm; it is the large round shape visible in the lower half of the image.

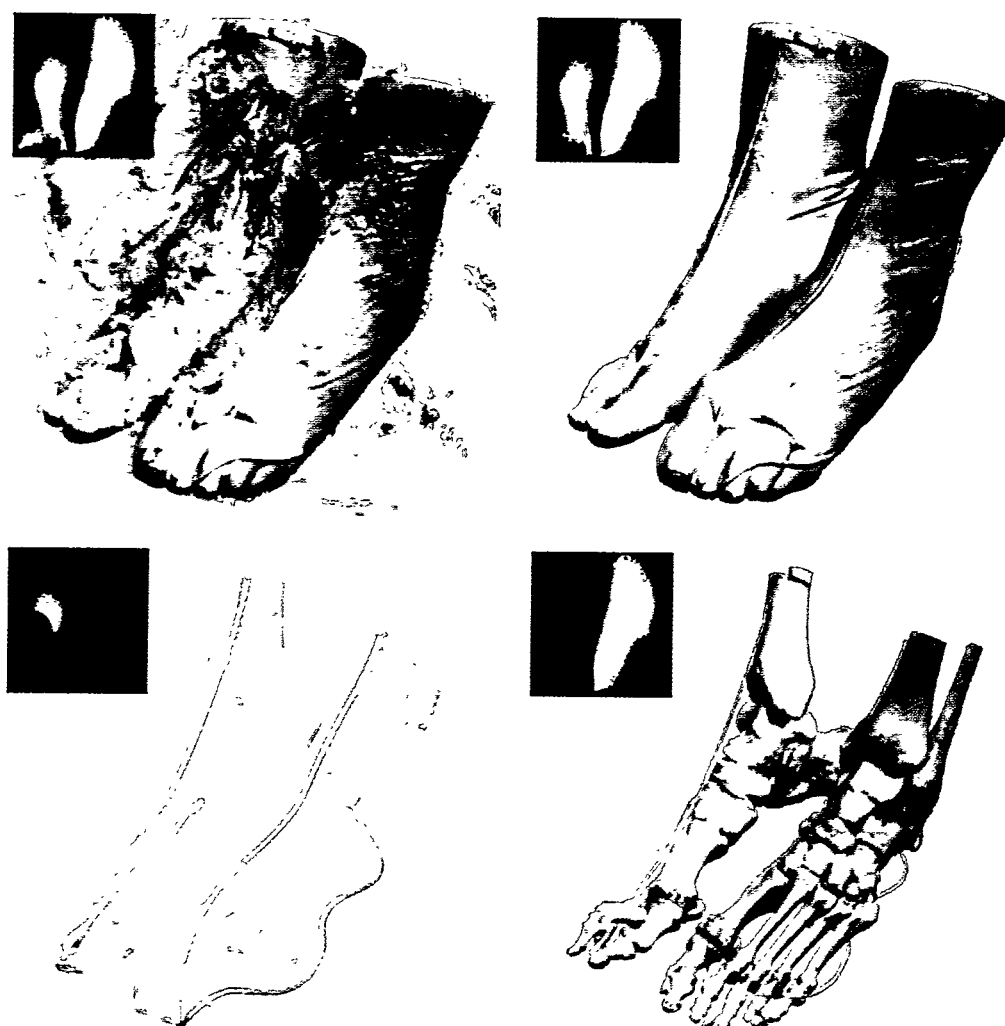


Figure 13: Renderings of the feet in the female Visible Human dataset. Inset in each rendered image is the two-dimensional opacity function used to generate it. At the upper-left is the initial automatically generated opacity function and rendering. Editing out a small region of opacity at low data value and low gradient magnitude removed the surrounding material from the rendering (upper-right). Careful selection in the opacity function allows imaging of the registration cord (lower-left). Finally, the bones are visualized by selecting the right-most portion of the opacity function (lower-right).

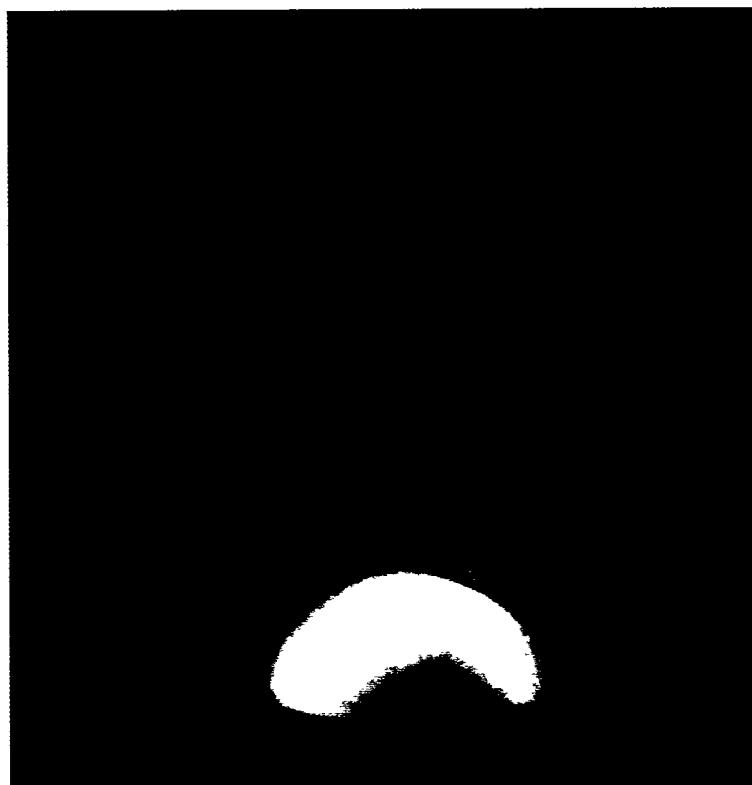


Figure 5: Image computed with XMPVO of a 13,000-cell complex.

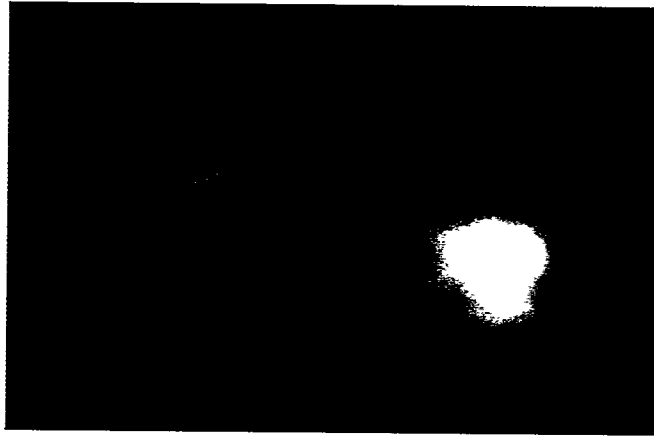


Figure 1: 5D interaction energy scalar field (Red=attraction, Blue=repulsion, Green=free movement). The axes configuration is reported on the bottom left (the stretched axis corresponds to a rotational degree of freedom).

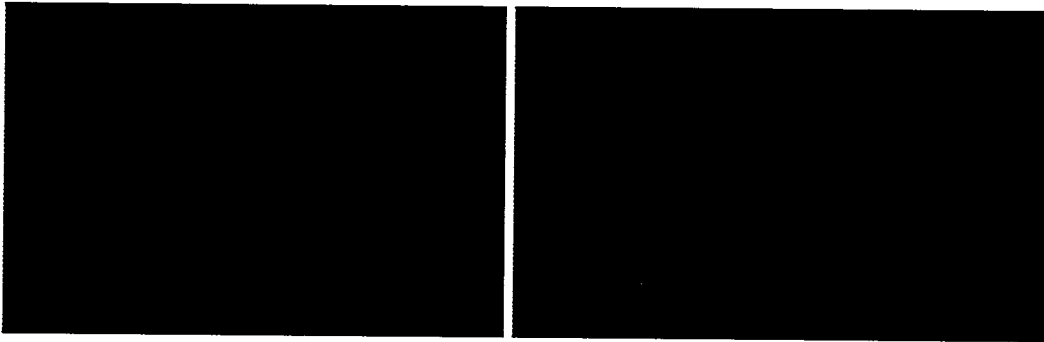


Figure 2: 5D interaction energy scalar field (Red=attraction, Blue=repulsion, Green=free movement). Same view as is figure 1 but highlighting only some of the energy components.

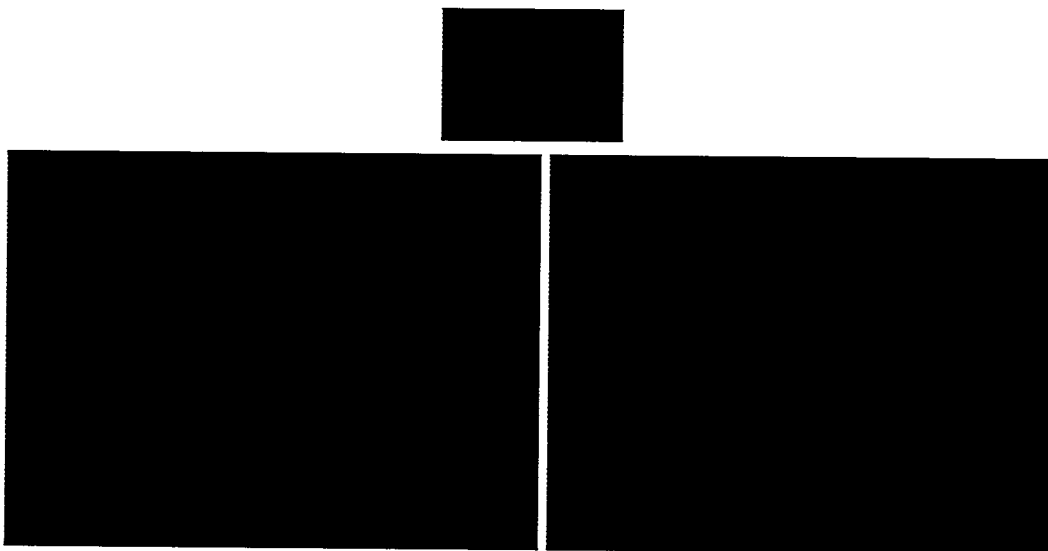
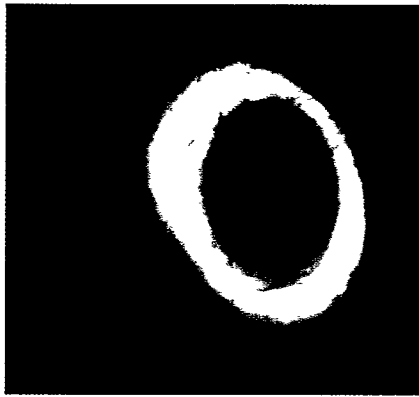
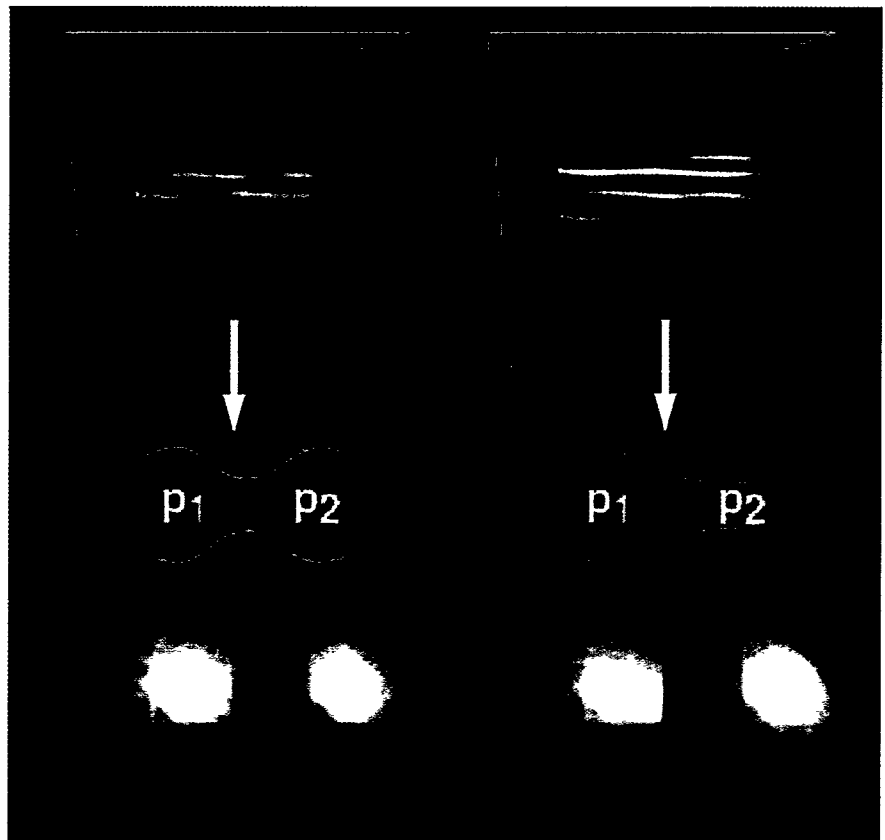
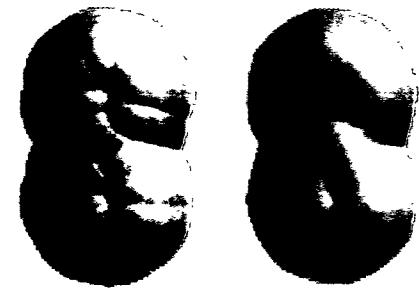
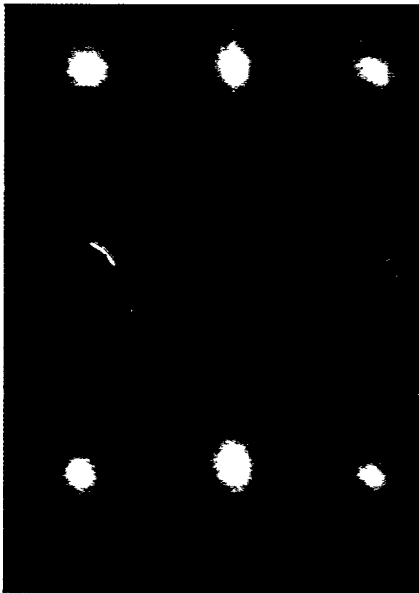


Figure 3: 5D interaction energy scalar field (Red=attraction, Blue=repulsion, Green=free movement). Same scalar field as in figure 1 but from a different view.



Weigle and Banks, "Extracting Iso-valued Features in 4-dimensional Scalar Fields"

CCW from upper-left. **Figure 3.** portion of an isovolume (wire frame) with an isosurface inside; volume swept by time-varying isosurface. **Figure 4.** Sphere-shaped isosurfaces sweeping a volume; the volume's envelope; dipole-antenna's envelope of iso field-strength. **Figure 6.** Blobby objects in higher dimensions projecting as interpenetrating objects. **Figure 7.** A family of isosurfaces at different resolutions within a single volumetric representation of a torus.



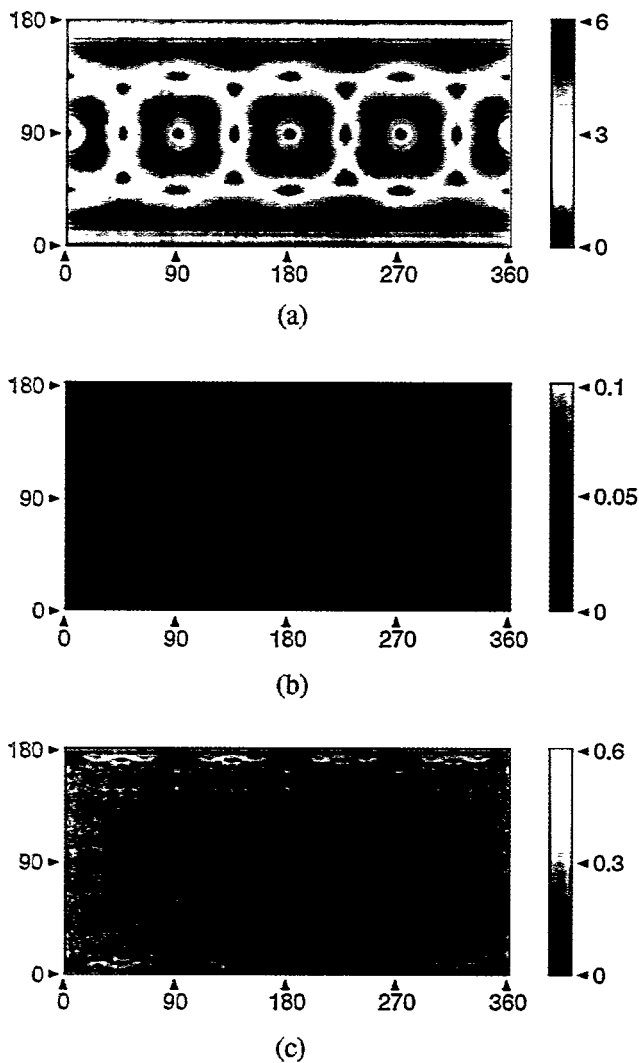


Figure 3: Experimental error estimation of the surface normal (in degrees) as a function of a sphere parametrization for (a) a Gaussian filter with $\sigma = 1$; (b) oriented box filter ($r_f = 1.8$) with float precision; and (c) same as (b) with unsigned char precision.

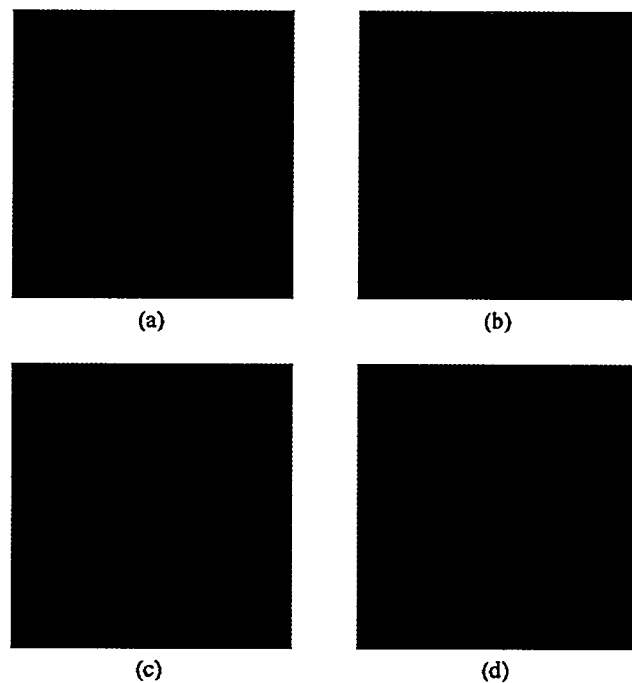


Figure 10: "Sharpening" of edges with increased raster resolution: (a) 64^3 , (b) 128^3 , (c) 256^3 , (d) 512^3 .

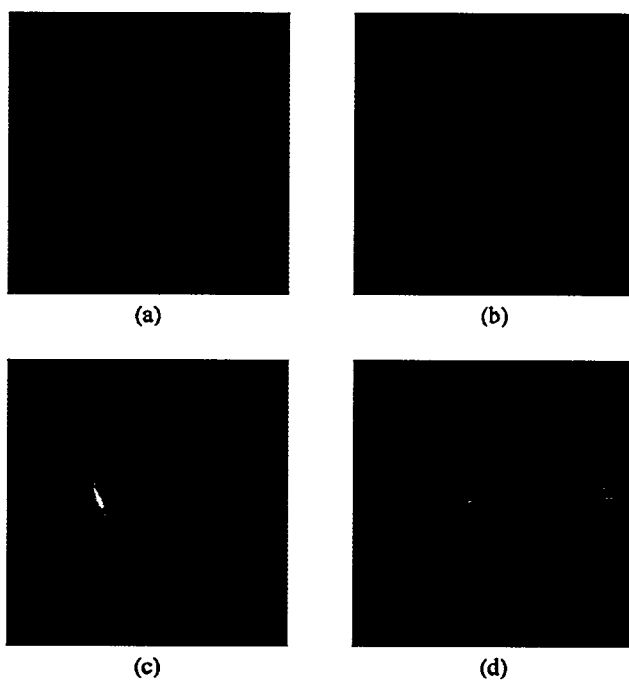


Figure 11: Voxelized parametric objects: (a) Monge patch, (b) Moebius strip, (c) ellipsoid, (d) Bezier patches.

Object Voxelization by Filtering
 Miloš Šrámek, Arie Kaufman

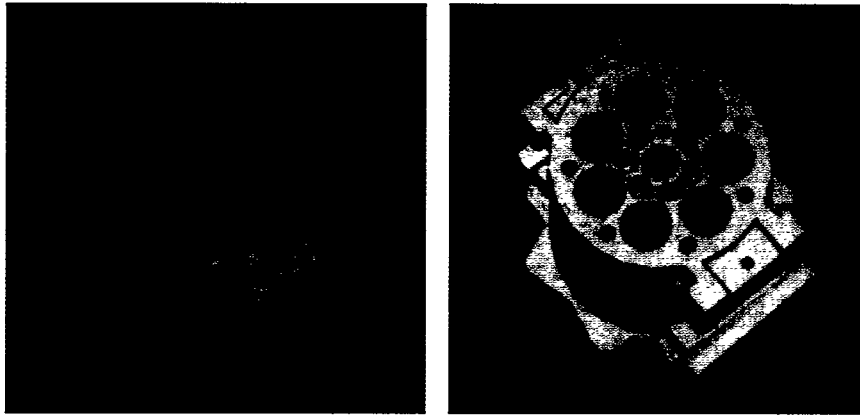


Figure 14. The 'Pump' data set voxelized in 26-separability (left) and 6-separability. They are almost identical visually, but are different topologically.

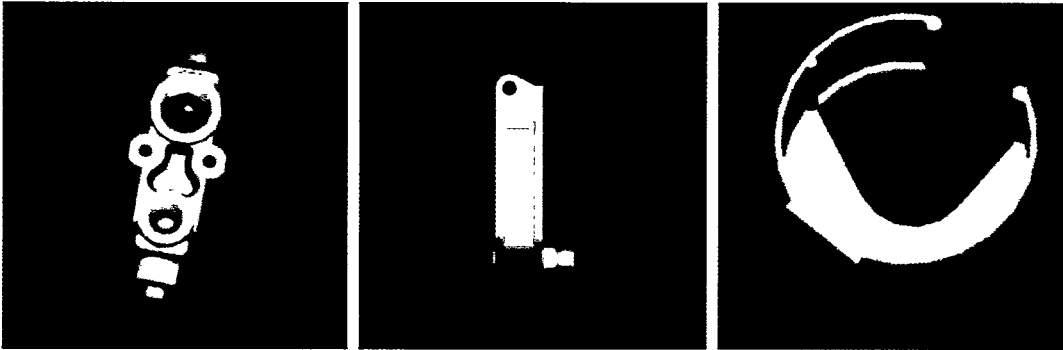


Figure 15. 6-separating voxelization of Bowl (right), Brevi (left), and Connector (middle).

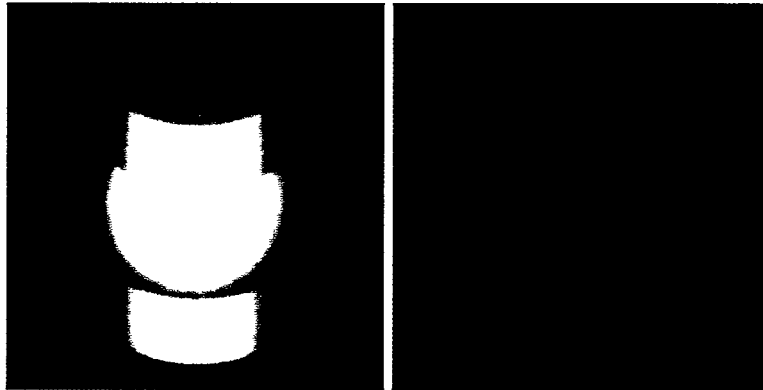


Figure 16. 6-separating voxelization of some cubic surfaces.

An Accurate Method To Voxelize Polygonal Meshes
Jian Huang, Roni Yagel, Vassily Filippov, Yair Kurzion

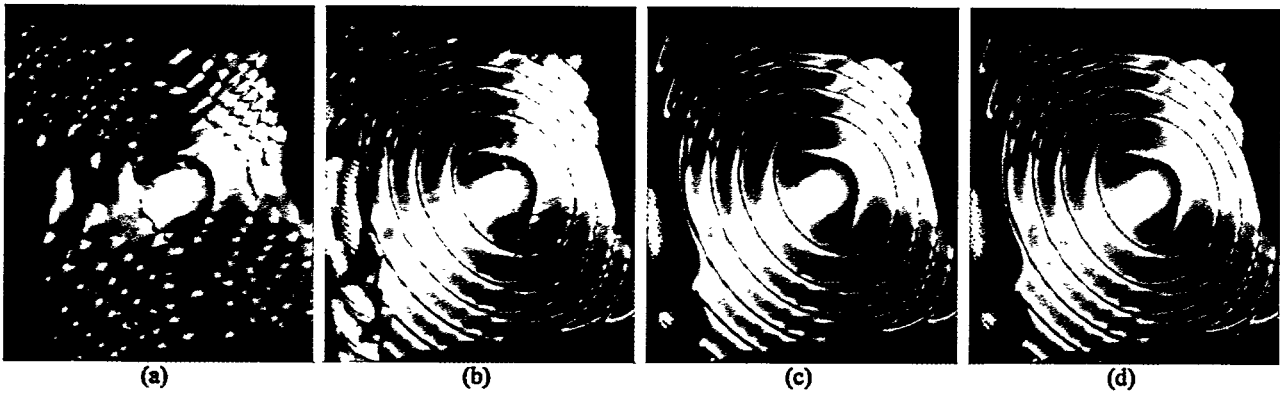


Figure 10: Image results of test function. (a) Linear (b) Adaptive $T_H=0, T_L=2$ (c) Adaptive $T_H=0, T_L=0$ (d) Spline.

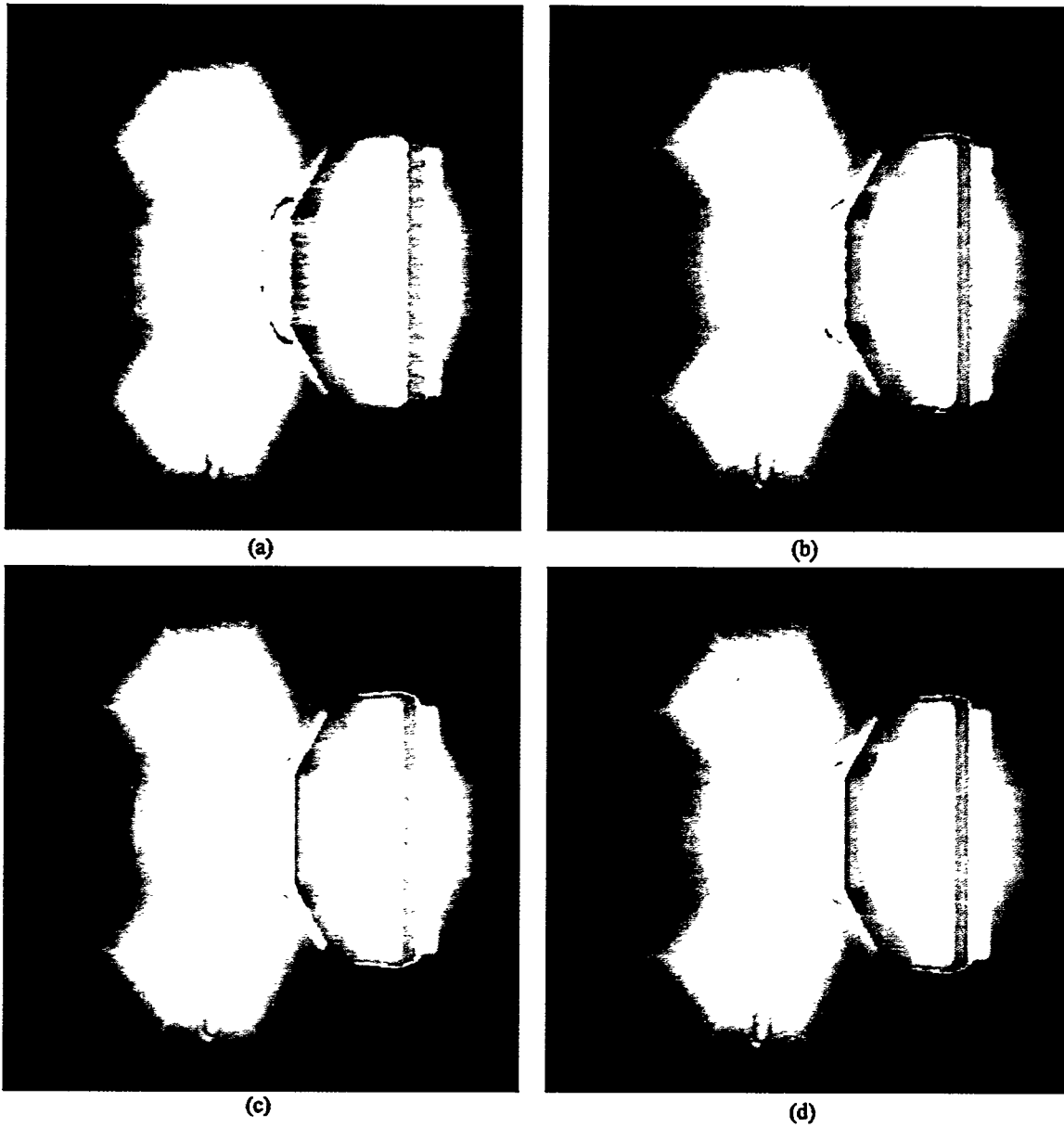


Figure 11: Image results of a bolt data set (direct volume rendering mixed with isosurfaces) (a) Linear (b) Adaptive $T_H=20, T_L=15$ (c) Adaptive $T_H=0, T_L=0$ (d) Spline.

Wavelet Based Adaptive Interpolation For Volume Rendering
Ricardo Sánchez, Marcelo Carvajal

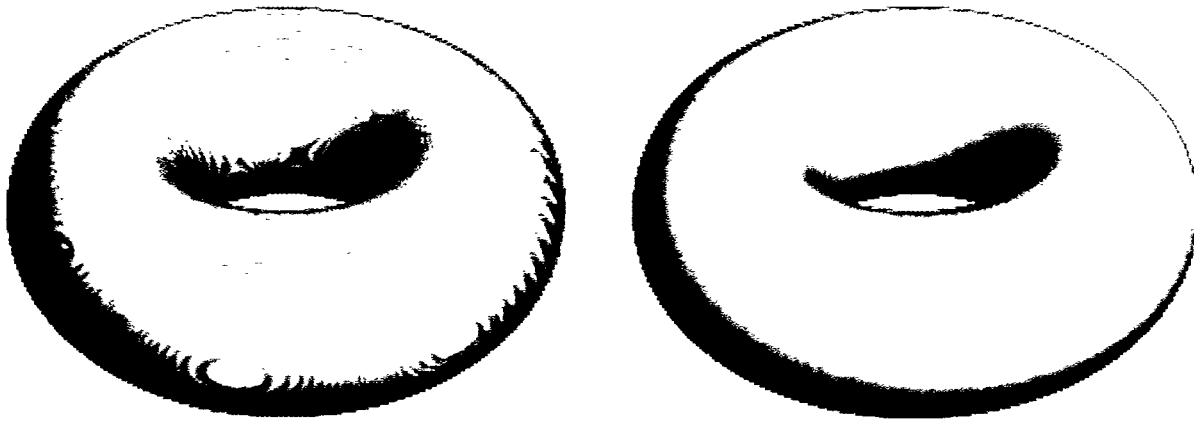


Figure 8. Left: Artifacts of separate interpolation of colors and opacity. Right: Improved using opacity-weighted color interpolation.



Figure 9. Left: Separate interpolation of color and opacity. Middle: Opacity-weighted interpolation of colors. Right: Normalized difference Image. Data courtesy of Dr. Ramani Pichumani, Stanford University.

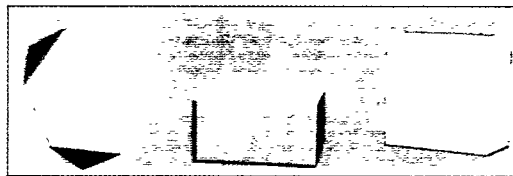


Figure 10. Three test rendering scenarios: corner, graze; and phase. One plane of the material boundaries has been antialiased during formation, by using a windowed sinc interpolation.



Figure 11. Left four images: from left, i) Artifacts of separate interpolation of colors, ii) opacity-weighted, iii) difference, iv) equalized difference. RMS Error 39.0. Right four images: same view and differences of antialiased face. RMS Error 17.7.



Figure 12. Left four images: from left, i) Artifacts of separate interpolation of colors, ii) opacity-weighted, iii) difference, iv) equalized difference. RMS Error 20.3. Right four images: same view and differences of antialiased face. RMS Error 8.32

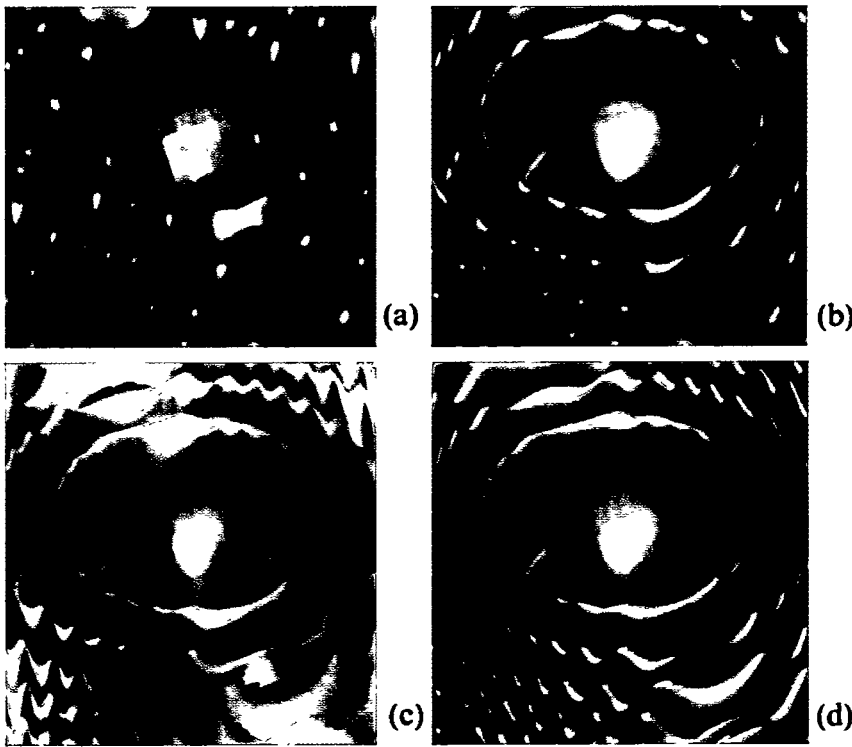


FIGURE 2. Marschner Lobb data set rendered using the following derivative filter (a) discontinuous 1EF (b) discontinuous 3EF (c) C^0 -1EF (d) C^0 -3EF (e) C^3 -4EF

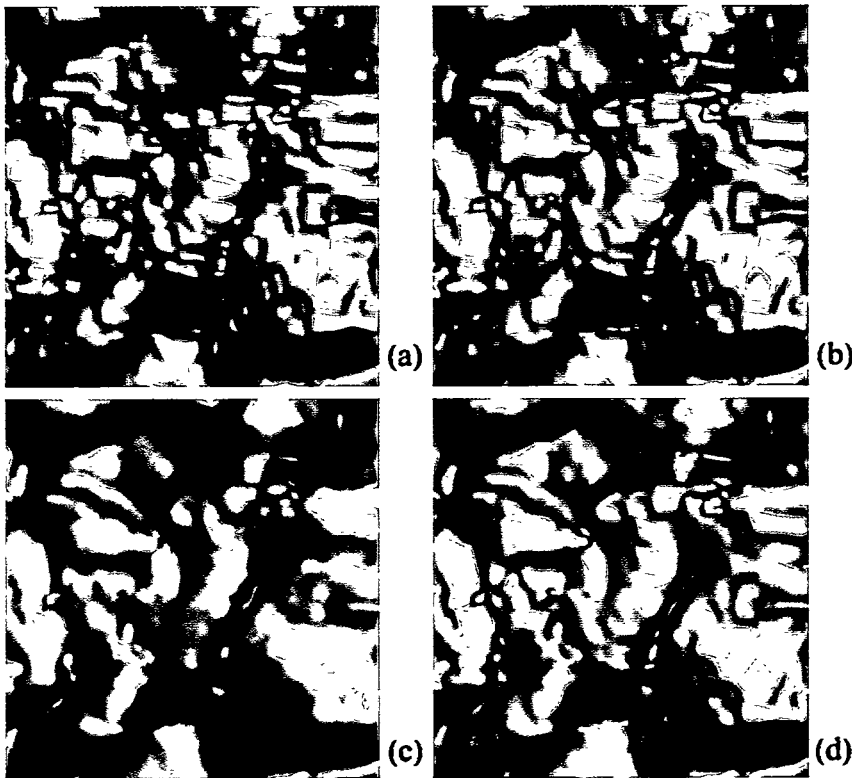


FIGURE 4. MRI data set rendered using the following derivative filter (a) discontinuous 1EF (b) discontinuous 3EF (c) C^0 -1EF (d) C^0 -3EF

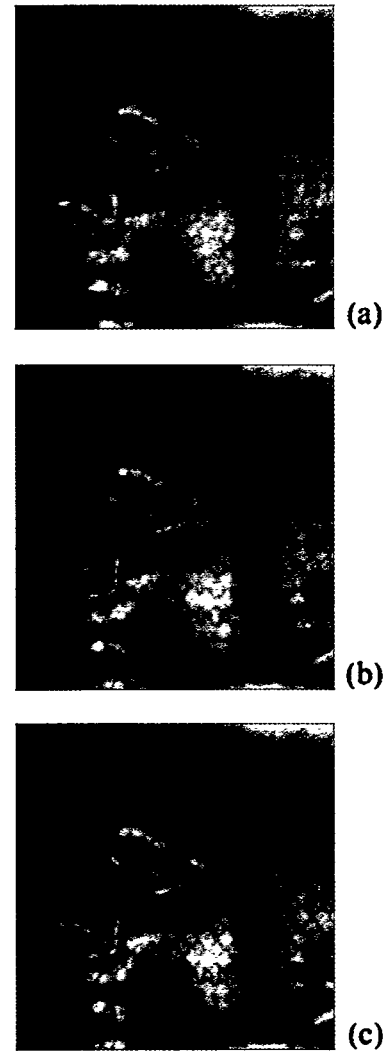


FIGURE 5. Size preserving pattern mapping of a texture on an MRI scan of a human head using (a) a discontinuous 1EF derivative filter (central differences) (b) a C^0 -2EF derivative filter (c) a C^1 -2EF derivative filter in order to determine the pattern density.

Design of Accurate And Smooth Filters For Function And Derivative Reconstruction
Torsten Möller, Klaus Mueller, Yair Kurzion, Raghu Machiraju, Roni Yagel